NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

AR MODELING OF COHERENCE IN
TIME DELAY AND DOPPLER ESTIMATION

by

Jun Lee

December 1988

Thesis Advisor Ralph Hippenstiel

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

1a Report Security Classification: Unclassified		1b Restrictive Markings	
2a Security Classification Authority		3 Distribution Availability of Report: Approved for public release; distribution is unlimited.	
2b Declassification Downgrading Schedule			
4 Performing Organization Report Number(s)		5 Monitoring Organization Report Number(s)	
6a Name of Performing Organization: Naval Postgraduate School	6b Office Symbol (*if applicable*): 62	7a Name of Monitoring Organization: Naval Postgraduate School	
6c Address (*city, state, and ZIP code*): Monterey, CA 93943-5000		7b Address (*city, state, and ZIP code*): Monterey, CA 93943-5000	
8a Name of Funding Sponsoring Organization	8b Office Symbol (*if applicable*)	9 Procurement Instrument Identification Number	
8c Address (*city, state, and ZIP code*)		10 Source of Funding Numbers	
		Program Element No	Project No / Task No / Work Unit Accession No

11 Title (*include security classification*)
AR MODELING OF COHERENCE IN TIME DELAY AND DOPPLER ESTIMATION

12 Personal Author(s): Jun Lee

13a Type of Report	13b Time Covered	14 Date of Report (*year, month, day*)	15 Page Count
Master's Thesis	From To	December 1988	89

16 Supplementary Notation: The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms (*continue on reverse if necessary and identify by block number*)
Field	Group	Subgroup	Coherence,Time delay and Doppler difference estimation

19 Abstract (*continue on reverse if necessary and identify by block number*)

The estimation of time delay and Doppler difference of a signal arriving at two physically separated sensors is investigated in this thesis. Usually, modified cross power spectrum coupled with Doppler compensation is used to detect a common, passive signal received at two separated sensors. Another successful approach uses the cross coherence to achieve this goal. This thesis modifies these two techniques to model the Doppler difference via an autoregressive (AR) technique. Analytical results are derived and experimentally verified via a computer simulation. Performance at high and low signal to noise ratios (*SNRs*) is examined.

20 Distribution Availability of Abstract ☒ unclassified unlimited ☐ same as report ☐ DTIC users	21 Abstract Security Classification Unclassified	
22a Name of Responsible Individual Ralph Hippenstiel	22b Telephone (*include Area code*) (408) 646-2633	22c Office Symbol 62Hi

DD FORM 1473,84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

Approved for public release; distribution is unlimited.

AR Modeling of Coherence in Time Delay and Doppler Estimation

by

Jun Lee
Captain, Korea Air Force
Korean Air Force Academy, 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1988

ABSTRACT

The estimation of time delay and Doppler difference of a signal arriving at two physically separated sensors is investigated in this thesis. Usually, modified cross power spectrum coupled with Doppler compensation is used to detect a common, passive signal received at two separated sensors. Another successful approach uses the cross coherence to achieve this goal. This thesis modifies these two techniques to model the Doppler difference via an autoregressive (AR) technique. Analytical results are derived and experimentally verified via a computer simulation. Performance at high and low signal to noise ratios (*SNRs*) is examined.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

I wish to gratefully express my appreciation to my thesis advisor, Professor Ralph Hippenstiel, for his invaluable efforts and patience in assisting me through my research. I also want to thank my wife Young Hi who made studying easier and our stay in Monterey very pleasant.

I. INTRODUCTION

This thesis investigates the use of autoregressive (AR) models for estimating the Doppler difference and differential time delay by processing of a narrow band signal emitted from a moving source and received at two physically separated sensors. If the signal is received at two different geographical positions even in the presence of uncorrelated noise, then, depending on the signal strength and duration, it is possible to estimate the differential time delay.

Because the source is moving, the signals that are received at the sensors may have different frequencies due to the Doppler effect. To obtain accurate differential time delay estimation, Doppler difference compensation is usually required.

This compensation can be implemented by using frequency shifting of the narrow band components of the received signal. This frequency shift can be obtained using a Fourier transform. In this thesis we use an AR model to detect the frequency shift. Using this Doppler compensation, an estimate of differential time delay can be obtained. Estimating the delay and Doppler using an AR model can be interpreted as a form of a narrow band coherence procedure, provided the estimations are properly normalized.

This thesis is arranged in six chapters and six appendices. Because the estimation of the time delay and Doppler is intimately related to the coherence between two transformed complex signals, an extensive investigation of coherence is given in Chapter II. In Chapter III, AR models, advantages of AR modeling, and AR model order selection are presented. Chapter IV is devoted to the analysis and the processing of noisy signals to estimate the differential time delay and the Doppler difference. To estimate the differential time delay, two approaches are pursued. AR model performance, Doppler estimation and two types of time delay estimation are examined in Chapter V. In the last chapter some general conclusions of the work carried out in this thesis are presented, and some suggestions for future investigations are given. Computer simulation programs are included in Appendices E and F.

II. COHERENCE

A. DEFINITION

The coefficient of coherence between two wide sense stationary random processes is the normalized cross power spectral density function defined by Wiener [Ref. 1: p. 12] as

$$\gamma_{xy}(f) = \frac{G_{xy}(f)}{\sqrt{G_{xx}(f)G_{yy}(f)}} \tag{2.1}$$

where f denotes the frequency (Hz) ,

$G_{xy}(f)$ is the cross power spectrum between $x(t)$ and $y(t)$,

$G_{xx}(f)$ denotes the auto power spectrum of $x(t)$, and

$G_{yy}(f)$ denotes the auto power spectrum of $y(t)$.

Despite some confusion in the literature, Wiener intended for the coefficient of coherence to be complex. This is apparent since he discusses both the modulus and the argument of the coefficient of coherence.

B. PROPERTY OF THE COHERENCE FUNCTION

The power spectral density matrix $Q(f)$ is positive semi definite. Therefore, for two random processes x and y, we see that

$$det\ [Q(f)] = det \begin{bmatrix} G_{xx}(f) & G_{xy}(f) \\ G_{yx}(f) & G_{yy}(f) \end{bmatrix} \geq 0 \tag{2.2}$$

For real processes we have $G_{yx}(f) = G^{*}_{xy}(f)$ and thus

$$G_{xx}(f)G_{yy}(f) - |G_{xy}(f)|^2 \geq 0 \tag{2.3}$$

where * denotes the conjugate of a complex number. And

$$G_{xx}(f)G_{yy}(f) \geq |G_{xy}|^2 \tag{2.4}$$

Furthermore, $G_{xx}(f)$ and $G_{yy}(f)$ are nonnegative, real functions of frequency. When $G_{xx}(f)$ and $G_{yy}(f)$ are strictly positive definite (i.e., $G_{xx}(f)G_{yy}(f) > 0$), Eq. (2.4) can be divided by $G_{xx}(f)G_{yy}(f)$ without changing the sense of the inequality.

This provides as an upper bound

$$|\gamma_{xy}(f)| \leq 1 \qquad \forall f \tag{2.5}$$

and since the magnitude of any complex number is greater than or equal to zero, we have the lower bound

$$0 \leq |\gamma_{xy}(f)| \leq 1 \tag{2.6}$$

The magnitude of the coherence function is always between zero and one. It is zero if the processes $x(t)$ and $y(t)$ are uncorrelated and it is equal to unity if there exists a linear relationship between $x(t)$ and $y(t)$. In order to define the coherence it is necessary that the numerator and denominator of Eq. (2.1) are not simultaneously equal to zero. Coherence is not defined if either auto spectra is zero.

C. COHERENCE ESTIMATION

If $X_l(f_k)$ denotes the Fast Fourier Transform (FFT) of the l th segment of $x(n)$ at frequency f_k, then the spectral density estimates are given by [Ref. 2: p. 22]

$$\hat{G}_{xx}(f_k) = \alpha \sum_{l=1}^{N} |X_l(f_k)|^2 \tag{2.7}$$

$$\hat{G}_{xy}(f_k) = \alpha \sum_{l=1}^{N} X_l(f_k) Y_l^*(f_k) \tag{2.8}$$

$$\hat{G}_{yy}(f_k) = \alpha \sum_{l=1}^{N} |Y_l(f_k)|^2 \tag{2.9}$$

where $\alpha = \dfrac{1}{NTf_s^2}$,

N = number of segments,

T = segment length, and

f_s = sampling frequency.

Finally the coherence estimation is

$$\hat{\gamma}(f_k) = \frac{\hat{G}_{xy}(f_k)}{\sqrt{\hat{G}_{xx}(f_k)\hat{G}_{yy}(f_k)}}$$
$$= \frac{\sum_{l=1}^{N} X_l(f_k) Y_l^*(f_k)}{\sqrt{\sum_{l=1}^{N} |X_l(f_k)|^2 \sum_{l=1}^{N} |Y_l(f_k)|^2}} \tag{2.10}$$

D. COHERENCE OF NARROW BAND SIGNALS WITH DIFFERENTIAL TIME DELAY AND DIFFERENTIAL DOPPLER

The output of the band pass filters (BPF1 and BPF2) in Figure 1 are denoted by $X_l(f_k)$ and $Y_l(f_k)$ respectively. Each term represents the Fourier transform of the corresponding time series evaluated at frequency f_k and time l. For narrow band signals a Doppler shift corresponds to a frequency shift. If a signal arrives at the two sensors having a differential Doppler shift as well as a differential time delay, then we see that a frequency compensation and time delay compensation by the appropriate values tend to line up the signals in frequency and time. This is accomplished by using an additional Fourier transform in channel-1 of the processor and a time delay in channel-2 of the processor. Mathematically, this can be expressed as

$$\hat{\gamma}(f_k) = \frac{\sum_{l=1}^{N} X_l(f_k) Y_{l+d}^*(f_k) e^{j\alpha l}}{\sqrt{\sum_{l=1}^{N} |X_l(f_k)|^2 \sum_{l=1}^{N} |Y_{l+d}(f_k)|^2}} \tag{2.11}$$

Comparing this with Eq. (2.10), we see that we generalized the coherence concept. We also note that the implementation resembles a correlator, where one of the signals is frequency compensated and the time delay corresponds to the delay operation of the correlator.

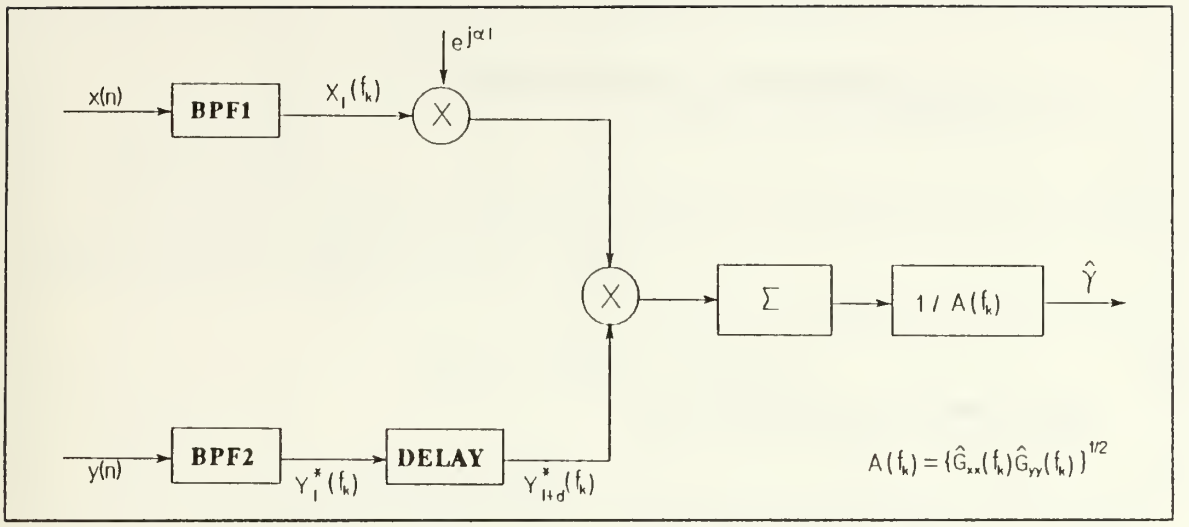

Figure 1. Coherence estimation block diagram.

If the Figure 1 is redrawn as in Figure 2, then it can be interpreted as an FFT implementation as shown in Figure 3.

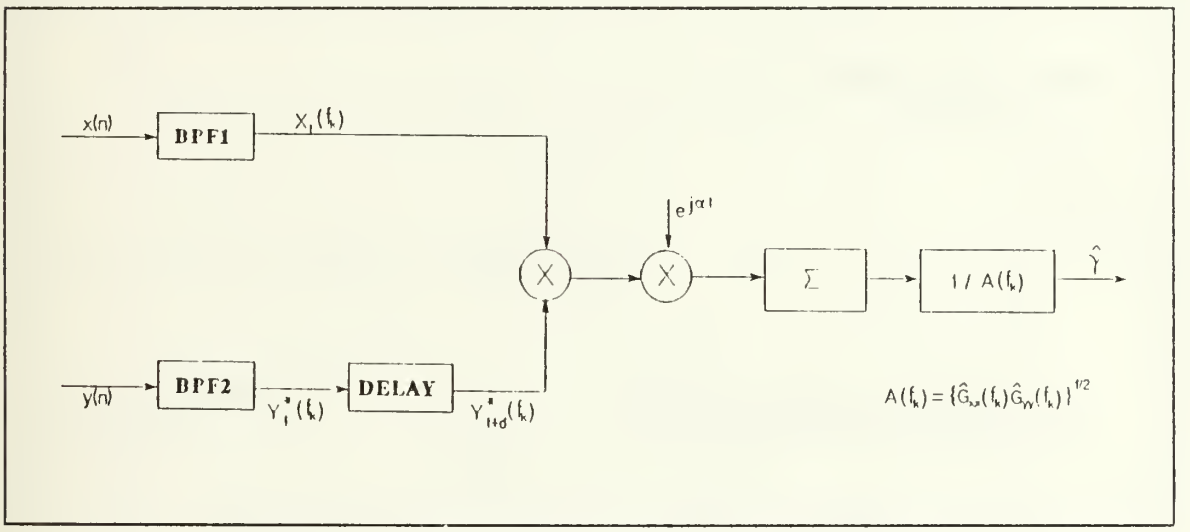

Figure 2. Coherence estimation block diagram (reinterpretation).

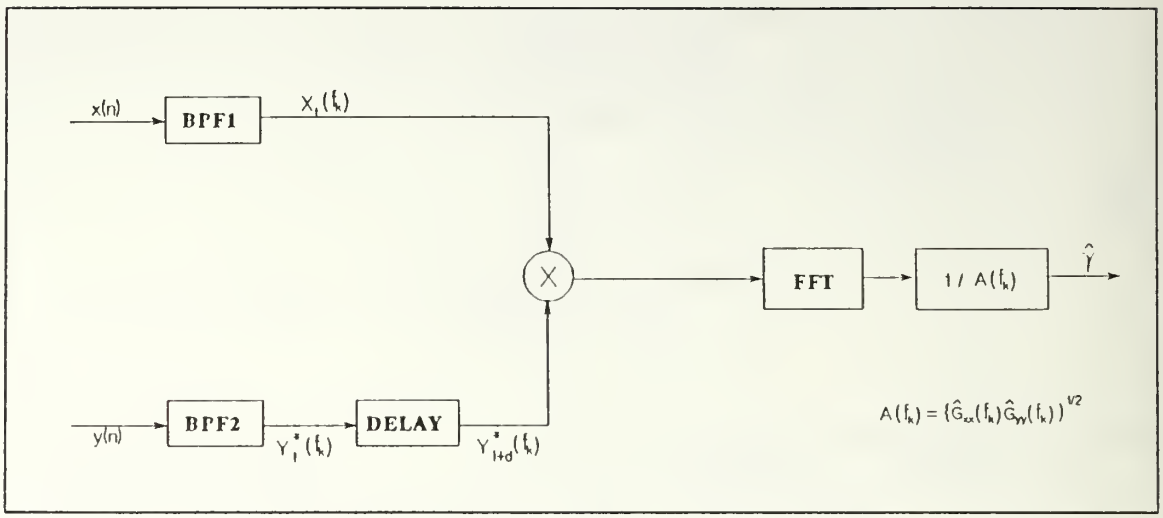

Figure 3. Coherence estimation block diagram using the FFT.

But this FFT has a poor resolution. To obtain a better resolution, an AR model is desirable. This approach is shown in Figure 4.

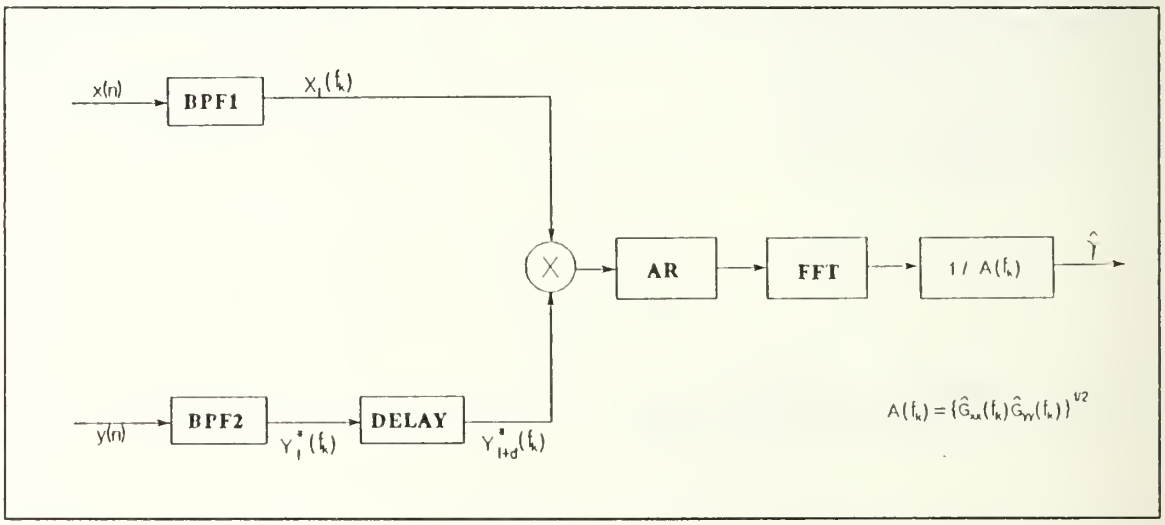

Figure 4. Coherence estimation block diagram using an AR model.

Figure 1 and Figure 4 represent two different schemes to compute the coherence function. Differential time delay and Doppler difference can be estimated by using AR models in the modeling of the coherence function.

III. AUTOREGRESSIVE (AR) MODELS

A. AR MODELING

If the following difference equation is satisfied, the resulting structure is called an AR model of order p. [Ref. 3: p. 177]

$$x(n) = -\sum_{k=1}^{p} a_k x(n-k) + u(n) \tag{3.1}$$

where $x(n)$ = the signal at instant n,

$u(n)$ = the white noise driver, and

a_k = the k th coefficient of the AR model.

A realization of the AR model is illustrated in Figure 5.

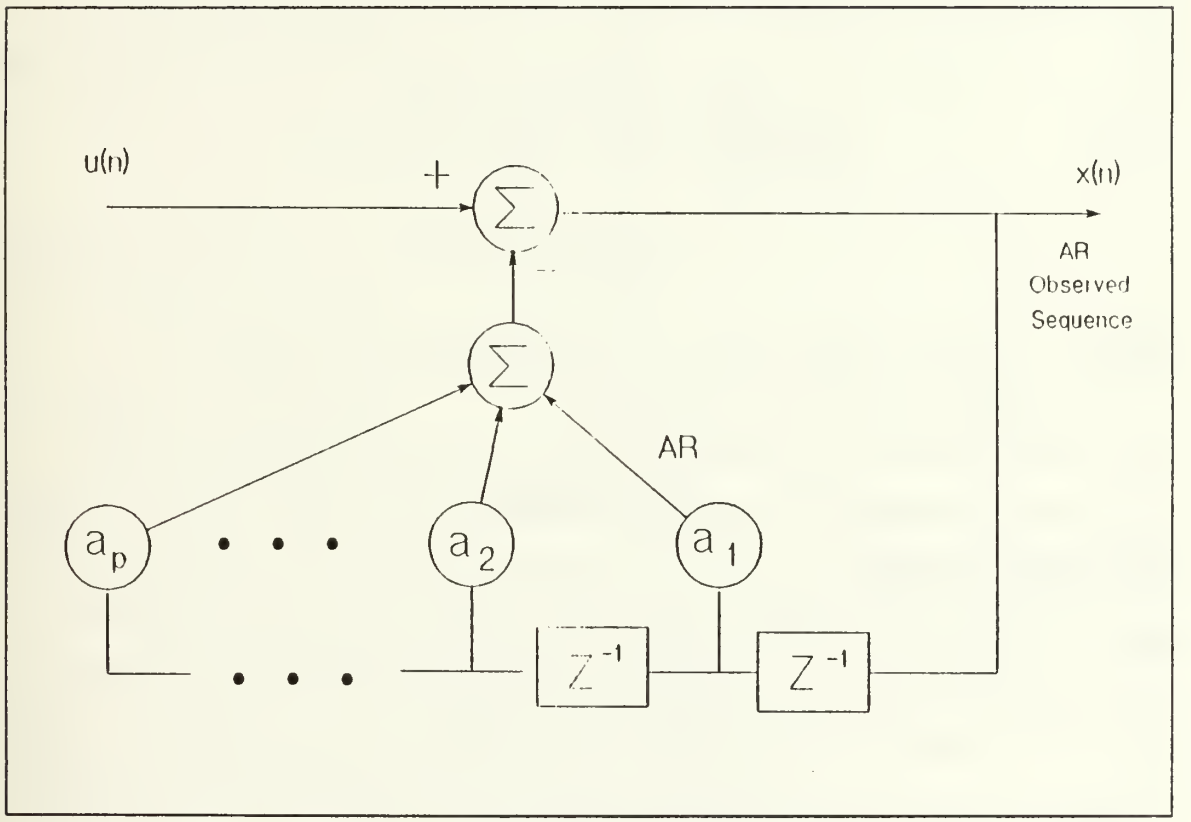

Figure 5. AR filter of order p.

B. ADVANTAGES OF THE AR MODEL

The motivation for parametric modeling of random processes is the ability to obtain better spectral estimates based upon the model than estimates produced by classical spectral estimation. Both the periodogram and correlation methods can be used to yield Power Spectral Density (PSD) estimates. The unavailable data or autocorrelation sequence (ACS) values outside a given window are assumed to be zero. This kind of an unrealistic assumption leads to distortions in the spectral estimate.

The advantages of the AR approach are

1. AR spectra tend to have sharp peaks, a desirable feature of high resolution spectral estimators.
2. AR parameter can be obtained as solutions to linear equations.

C. POWER SPECTRAL DENSITY

Eq. (3.1) can be rewritten as follows

$$x(n) = -\sum_{k=1}^{p} a_k x(n-k) + u(n)$$

$$= \sum_{k=0}^{\infty} h_k u(n-k) \tag{3.2}$$

where h_k = the causal filter impulse response.

Let us take the Z -transform of Eq(3.2).

$$X(z) = H(z)U(z) \tag{3.3}$$

Rewriting Eq. (3.1) as

$$\sum_{k=0}^{p} a_k x(n-k) = u(n) \tag{3.4}$$

where $a_0 = 1$, and taking the Z -transform gives

$$A(z)X(z) = U(z) \tag{3.5}$$

Eq. (3.3) and Eq. (3.5) can be solved to obtain H(z)

$$H(z) = \frac{1}{A(z)} \tag{3.6}$$

and hence

$$X(z) = \frac{U(z)}{A(z)} \tag{3.7}$$

The Z - transform of the output sequence $\{x(n)\}$ is related to the Z - transformation of the input random process $u(n)$ by [Ref. 4: p. 56]

$$P_{xx}(z) = X(z)X^*\left(\frac{1}{z^*}\right) = \frac{U(z)U^*\left(\frac{1}{z^*}\right)}{A(z)A^*\left(\frac{1}{z^*}\right)} = P_{uu}(z)\frac{1}{A(z)A^*\left(\frac{1}{z^*}\right)} \tag{3.8}$$

The AR power spectral density is obtained by substituting $z = e^{j2\pi fT}$ into Eq. (3.8) and scaling it by the interval T.

$$P_{AR}(f) = T\rho_\omega \frac{1}{|A(f)|^2} = T\rho_\omega \frac{1}{e_p^H(f)aa^H e_p(f)} \tag{3.9}$$

where $e_p = [1, e^{-j2\pi fT}, \ldots, e^{-j2\pi fpT}]^H$

$a = [1, a_1, a_2, \ldots, a_p]^T$

ρ_ω = variance of driving sequence.

D. BURG'S ALGORITHM

In practice, the autocorrelation is usually not available, so one must make an AR spectral estimation based on the available data samples. The simplest procedure to obtain an AR spectral estimate from data samples would be to produce estimates of the autocorrelation sequence from the data. These autocorrelation estimates would be used in lieu of the true autocorrelation sequence in the YULE - WALKER equations to yield the AR coefficients. However better results are obtained, particulary for short data segments, by algorithms that obtain the AR model parameters directly from the data, without explicitly estimating the autocorrelation function.

The Levinson recursive solution to the YULE - WALKER equations relates the pth order AR parameters to the $p-1$ th order parameters as given by [Ref. 3: p. 211]

$$a_p(n) = a_{p-1}(n) + k_p a_{p-1}^*(p-n) \tag{3.10}$$

For $n = 1$ to $n = p - 1$. the reflection coefficients k_p can be found by using the known autocorrelation function for lag 0 to $p - 1$. So we have

$$k_p = a_p(p) = \frac{-\sum_{n=0}^{p-1} a_{p-1}(n) r_{xx}(p-n)}{\rho_{p-1}} \tag{3.11}$$

The recursion for the driving white noise variance is given by

$$\rho_p = \rho_{p-1}(1 - |k_p|^2) \tag{3.12}$$

where $\rho_0 = r_{xx}(0)$.

But the ACS is not available, hence we can not calculate the reflection coefficients. The Burg algorithm provides an estimate of the reflection coefficients which in turm are obtained through a least squares criterion.

The forward linear prediction error is given by

$$e_p^f(n) = x(n) + \sum_{m=1}^{p} a_p^f(m) x(n-m) \tag{3.13}$$

while the backward linear prediction error is given by

$$e_p^b(n) = x(n-p) + \sum_{m=1}^{p} a_p^{f*}(m) x(n+m-p) \tag{3.14}$$

Substitution of Eq. (3.10) into Eq. (3.13) and Eq. (3.14) yields the recursive relationships

$$e_p^f(n) = e_{p-1}^f(n) + k_p e_{p-1}^b(n-1) \tag{3.15}$$

$$e_p^b(n) = e_{p-1}^b(n-1) + k_p^* e_{p-1}^f(n) \tag{3.16}$$

At each order p, the arithmetic mean of the forward and backward linear prediction error power (sample prediction error variance) is given by

$$\rho_p^{fb} = \frac{1}{2}\left[\frac{1}{N}\sum_{n=p+1}^{N} |e_p^f(n)|^2 + \frac{1}{N}\sum_{n=p+1}^{N} |e_p^b(n)|^2\right] \tag{3.17}$$

This expression is minimized, subject to the recursion given by Eq. (3.15) and Eq. (3.16). Thus, ρ_p^{fb} is a function of single parameter, namely the complex valued reflection coefficient k_p. Setting the complex derivative of Eq. (3.17) to zero

$$\frac{\partial \rho_b^{fb}}{\partial Re(k_p)} + j\frac{\partial \rho_b^{fb}}{\partial Im(k_p)} = 0 \tag{3.18}$$

and solving for k_p yields

$$\hat{k}_p = \frac{-2\sum_{n=p+1}^{N} e_{p-1}^f(n) e_{p-1}^{b\,*}(n-1)}{\sum_{n=p+1}^{N} |e_{p-1}^f(n)|^2 + \sum_{n=p+1}^{N} |e_{p-1}^b(n-1)^2} \tag{3.19}$$

The estimation of the reflection coefficient represents the HARMONIC mean between the forward and backward partial correlation coefficient, where $e_0^f(n) = e_0^b(n) = x(n)$ and N = number of data points.

E. FINAL PREDICTION ERROR (FPE) CRITERION

Because the best choice of filter order is not generally known a priori, it is usually necessary in practice to postulate several model orders. FPE is a kind of criterion which was provided by Akaike [Ref. 3: p. 230]. This criterion selects the order of the AR process so that the average error variance equals the sum of the power in the unpredictable part of the process and of a quantity representing the inaccuracies in estimating the AR parameters. The FPE for an AR process is defined as

$$FPE(p) = \hat{\rho}_p \frac{N + (p+1)}{N - (p+1)} \tag{3.20}$$

where N is the number of data samples,

p is order of the filter, and

$\hat{\rho}_p$ is the estimated white noise variance when using a p th order filter.

The order p selected is the one for which the FPE is minimum.

IV. DOPPLER AND DIFFERENTIAL TIME DELAY ESTIMATION

A. DOPPLER ESTIMATION

Let $x(t)$ and $y(t)$ be the signals received by two sensors

$$x(t) = \cos\{2\pi(f + \alpha_1)t\} \tag{4.1}$$

$$y(t) = \cos\{2\pi(f + \alpha_2)t\} \tag{4.2}$$

where f is the carrier frequency,

α_1 , α_2 are Doppler shifts, and

t is the time variable.

Let f_s be the sampling frequency which satisfies the Nyquist theorem. For convenience, in all derivations and simulations a sampling frequency of 64 Hz is used, together with band pass filter width of 1 Hz. We assume that $|\alpha_i| < 0.5$ ($i = 1,2$) , which implies that the signals stay in the band pass filter regions of their respective band pass filters regardless of any Doppler shift. Note, these values can be modified to arbitrary sampling rates and pass band regions. The sampling rate is given by

$$f_s \geq 2f + 1 > 2(f + \alpha_i) \qquad (\ i = 1,2\ ) \tag{4.3}$$

The phase at instant k and sampling time interval T are given by

$$_x\phi_k = \frac{2\pi(f + \alpha_1)k}{f_s} \tag{4.4}$$

$$_y\phi_k = \frac{2\pi(f + \alpha_2)k}{f_s} \tag{4.5}$$

$$T = \frac{1}{f_s} \tag{4.6}$$

Let $x(n)$ denote the sampled analog signal $x(t)|_{t=nT}$ then

$$x(n + m) = \cos\{2\pi(f + \alpha_1)\frac{n}{f_s} + {}_x\phi_m\} \tag{4.7}$$

$$y(n+m) = \cos\{2\pi(f+\alpha_2)\frac{n}{f_s} + {}_y\phi_m\} \tag{4.8}$$

The derivations of Eq. (4.7) and Eq. (4.8) are given in Appendix A.

These signals, $\{x(n)\}$ and $\{y(n)\}$, are processed to detect the Doppler difference . Let BPF1 be the band pass filter centered at f_1 and BPF2 be the band pass filter centered at f_2 (where f_2 might equal f_1). N data points are processed in the band pass filters, to generate one output. The inputs to BPF1 and BPF2 at time m are vectors $X_m(n)$ and $Y_m(n)$, respectively. The size of the input vector to the filter is the number of data points taken during 1 second (i.e., the number of points processed in the filter = $N = f_s$). The input vectors are denoted by

$$X_m(n) = [x(m), x(m+1), \ldots\ldots, x(m+N-1)] \tag{4.9}$$

$$Y_m(n) = [y(m), y(m+1), \ldots\ldots, y(m+N-1)] \tag{4.10}$$

The filtering is performed using FFTs , where successive FFT outputs are generated at the input data rate. The BPF2 output is conjugated.

To avoid the complexity the following four complex constant variables are defined.

$$A_x = \sum_{n=0}^{N-1} \cos\{2\pi(f+\alpha_1)\frac{n}{f_s}\} e^{-j2\pi f\frac{n}{N}} \tag{4.11}$$

$$B_x = \sum_{n=0}^{N-1} \sin\{2\pi(f+\alpha_1)\frac{n}{f_s}\} e^{-j2\pi f\frac{n}{N}} \tag{4.12}$$

$$A_y = \sum_{n=0}^{N-1} \cos\{2\pi(f+\alpha_2)\frac{n}{f_s}\} e^{j2\pi f\frac{n}{N}} \tag{4.13}$$

$$B_y = \sum_{n=0}^{N-1} \sin\{2\pi(f+\alpha_2)\frac{n}{f_s}\} e^{j2\pi f\frac{n}{N}} \tag{4.14}$$

At instant m the complex output signal of BPF1 can be calculated as follows

$$
\begin{aligned}
X_m(f) &= \sum_{n=0}^{N-1} \cos\{2\pi(f+\alpha_1)\frac{n}{f_s} + {}_x\phi_m\} e^{-j2\pi f\frac{n}{N}} \\
&= \sum_{n=0}^{N-1} [\cos\{2\pi(f+\alpha_1)\frac{n}{f_s}\}\cos {}_x\phi_m - \sin\{2\pi(f+\alpha_1)\frac{n}{f_s}\}\sin {}_x\phi_m] e^{-j2\pi f\frac{n}{N}} \\
&= \cos {}_x\phi_m \sum_{n=0}^{N-1} \cos\{2\pi(f+\alpha_1)\frac{n}{f_s}\} e^{-j2\pi\frac{n}{N}} - \sin {}_x\phi_m \sum_{n=0}^{N-1} \sin\{2\pi(f+\alpha_1)\frac{n}{f_s}\} e^{-j2\pi\frac{n}{N}} \\
&= A_x \cos {}_x\phi_m - B_x \sin {}_x\phi_m \\
&= A_x \frac{e^{j_x\phi_m} + e^{-j_x\phi_m}}{2} - B_x \frac{e^{j_x\phi_m} - e^{-j_x\phi_m}}{2j} \\
&= \frac{A_x + jB_x}{2} e^{j_x\phi_m} + \frac{A_x - jB_x}{2} e^{-j_x\phi_m} \\
&= \frac{A_x + jB_x}{2} e^{j2\phi\frac{(f+\alpha_1)}{f_s}m} + \frac{A_x - jB_x}{2} e^{-j2\phi\frac{(f+\alpha_1)}{f_s}m}
\end{aligned}
\tag{4.15}
$$

The complex signal $X_m(f)$ contains two frequencies with different amplitudes.

To understand the character of $X_m(f)$, it is important to evaluate which is the dominant term in the above expression.

$$
\begin{aligned}
A_x + jB_x &= \sum_{n=0}^{N-1} [\cos\{2\pi(f+\alpha_1)\frac{n}{f_s}\} + j\sin\{2\pi(f+\alpha_1)\frac{n}{f_s}\}] e^{-j2\pi f\frac{n}{N}} \\
&= \sum_{n=0}^{N-1} e^{j2\pi(f+\alpha_1)\frac{n}{f_s}} e^{-j2\pi f\frac{n}{N}} \\
&= \sum_{n=0}^{N-1} e^{j2\pi\alpha_1\frac{n}{N}} \\
&= \frac{1 - e^{j2\pi\alpha_1}}{1 - e^{j2\pi\alpha_1\frac{1}{N}}}
\end{aligned}
\tag{4.16}
$$

$$
\begin{aligned}
A_x - jB_x &= \sum_{n=0}^{N-1} [\cos\{2\pi(f+\alpha_1)\frac{n}{f_s}\} - j\sin\{2\pi(f+\alpha_1)\frac{n}{f_s}\}] e^{-j2\pi f\frac{n}{N}} \\
&= \sum_{n=0}^{N-1} e^{-j2\pi(f+\alpha_1)\frac{n}{f_s}} e^{-j2\pi f\frac{n}{N}} \\
&= \sum_{n=0}^{N-1} e^{-j2\pi(2f+\alpha_1)\frac{n}{N}} \\
&= \frac{1-e^{-j2\pi\alpha_1}}{1-e^{-j2\pi(2f+\alpha_1)\frac{1}{N}}}
\end{aligned}
\tag{4.17}
$$

From Appendix C

$$
|A_x + jB_x| > |A_x - jB_x| \tag{4.18}
$$

Therefore $\frac{A_x + jB_x}{2} e^{j\alpha_m}$ is the dominant term of $X_m(f)$.

In similiar way, the complex output signal of BPF2 can be calculated as follows

$$
\begin{aligned}
Y_m^*(f) &= \sum_{n=0}^{N-1} \cos\{2\pi(f+\alpha_2)\frac{n}{f_s} + {}_y\phi_m\} e^{j2\pi f\frac{n}{N}} \\
&= \sum_{n=0}^{N-1} [\cos\{2\pi(f+\alpha_2)\frac{n}{f_s}\}\cos {}_y\phi_m - \sin\{2\pi(f+\alpha_2)\frac{n}{f_s}\}\sin {}_y\phi_m] e^{j2\pi f\frac{n}{N}} \\
&= \cos {}_y\phi_m \sum_{n=0}^{N-1} \cos\{2\pi(f+\alpha_2)\frac{n}{f_s}\} e^{j2\pi\frac{n}{N}} - \sin {}_y\phi_m \sum_{n=0}^{N-1} \sin\{2\pi(f+\alpha_2)\frac{n}{f_s}\} e^{-j2\pi\frac{n}{N}} \\
&= A_y \cos {}_y\phi_m - B_y \sin {}_y\phi_m \\
&= A_y \frac{e^{j_y\phi_m} + e^{-j_y\phi_m}}{2} - B_y \frac{e^{j_y\phi_m} - e^{-j_y\phi_m}}{2j} \\
&= \frac{A_y + jB_y}{2} e^{j_y\phi_m} + \frac{A_y - jB_y}{2} e^{-j_y\phi_m} \\
&= \frac{A_y + jB_y}{2} e^{j2\phi\frac{(f+\alpha_2)}{f_s}m} + \frac{A_y - jB_y}{2} e^{-j2\phi\frac{(f+\alpha_2)}{f_s}m}
\end{aligned}
\tag{4.19}
$$

The complex signal $Y_m^*(f)$ contains two frequencies with different amplitudes.

Even though $Y_m^*(f)$ is similiar in form to $X_m(f)$, it is not obvious which term is dominant.

$$
\begin{aligned}
A_y + jB_y &= \sum_{n=0}^{N-1} [\cos\{2\pi(f+\alpha_2)\frac{n}{f_s}\} + j\sin\{2\pi(f+\alpha_2)\frac{n}{f_s}\}] e^{j2\pi f\frac{n}{N}} \\
&= \sum_{n=0}^{N-1} e^{j2\pi(f+\alpha_2)\frac{n}{f_s}} e^{j2\pi f\frac{n}{N}} \\
&= \sum_{n=0}^{N-1} e^{j2\pi(2f+\alpha_2)\frac{n}{N}} \\
&= \frac{1-e^{j2\pi(2f+\alpha_2)}}{1-e^{j2\pi(2f+\alpha_2)\frac{1}{N}}} \\
&= \frac{1-e^{j2\pi\alpha_2}}{1-e^{j2\pi(2f+\alpha_2)\frac{1}{N}}}
\end{aligned}
\tag{4.20}
$$

$$
\begin{aligned}
A_y - jB_y &= \sum_{n=0}^{N-1} [\cos\{2\pi(f+\alpha_2)\frac{n}{f_s}\} - j\sin\{2\pi(f+\alpha_2)\frac{n}{f_s}\}] e^{j2\pi f\frac{n}{N}} \\
&= \sum_{n=0}^{N-1} e^{-j2\pi(f+\alpha_2)\frac{n}{f_s}} e^{j2\pi f\frac{n}{N}} \\
&= \sum_{n=0}^{N-1} e^{-j2\pi\alpha_2\frac{n}{N}} \\
&= \frac{1-e^{-j2\pi\alpha_2}}{1-e^{-j2\pi\alpha_2\frac{1}{N}}}
\end{aligned}
\tag{4.21}
$$

From Appendix C

$$|A_y - jB_y| > |A_y + jB_y| \tag{4.22}$$

Therefore $\frac{A_y - jB_y}{2} e^{j\varphi_m}$ is the dominant term of $Y_m^*(f)$.

Since the two output sequences $\{X_m(f)\}$ and $\{Y_m(f)\}$ have two sinusoidal components each, their product $\{X_m(f)Y_m^*(f)\}$ has four sinusoidal components. The four frequencies are

$$\omega_1 = 2\pi\frac{(\alpha_1-\alpha_2)}{f_s}\ ,\ \omega_2 = 2\pi\frac{(\alpha_2-\alpha_1)}{f_s}\ ,\ \omega_3 = 2\pi\frac{(2f+\alpha_1+\alpha_2)}{f_s}\ ,\ \omega_4 = -2\pi\frac{(2f+\alpha_1+\alpha_2)}{f_s}$$

with ω_1 the sinusoid with the largest amplitude and ω_2 the sinusoid with the smallest amplitude.

The product of the output of the filters is given by

$$\begin{aligned} X_m(f)Y_m^*(f) &= \frac{A_x+jB_x}{2}e^{j_x\phi_m}\frac{A_y-jB_y}{2}\frac{e^{-j_y\phi_m}}{2}+F(\alpha_1,\alpha_2,{}_x\phi_m,{}_y\phi_m) \\ &= \frac{(A_x+jB_x)(A_y-jB_y)}{4}e^{j({}_x\phi_m-{}_y\phi_m)}+F(\alpha_1,\alpha_2,{}_x\phi_m,{}_y\phi_m) \end{aligned} \tag{4.23}$$

where $F(\alpha_1,\alpha_2,{}_x\phi_m,{}_y\phi_m)$ represents the three low amplitude frequency terms.

When using the AR model as described in chapter III, at a high SNR (i.e., $SNR\rightarrow\infty$) , a 4th order AR model detects all of the four frequencies given above. When the SNR is low (i.e., 20 dB), only one dominant sinusoid is detected with frequency

$$\begin{aligned} \omega_1 &= \frac{2\pi(f+\alpha_1)}{f_s}-\frac{2\pi(f+\alpha_2)}{f_s} \\ &= 2\pi\frac{(\alpha_1-\alpha_2)}{f_s} \end{aligned} \tag{4.24}$$

Since we can detect ω_1 and know the value of f_s , $\alpha_1-\alpha_2$ can be estimated by using an AR model.

B. DIFFERENTIAL TIME DELAY ESTIMATION

To detect and localize a target, it is important to find the differential time delay of signals arriving at two sensors. Let us assume that signals at the two sensors are as given in Figure 6 (i.e., zero differential time delay) For the remainder of the figures, the time axis is scaled to be 64 points per second.

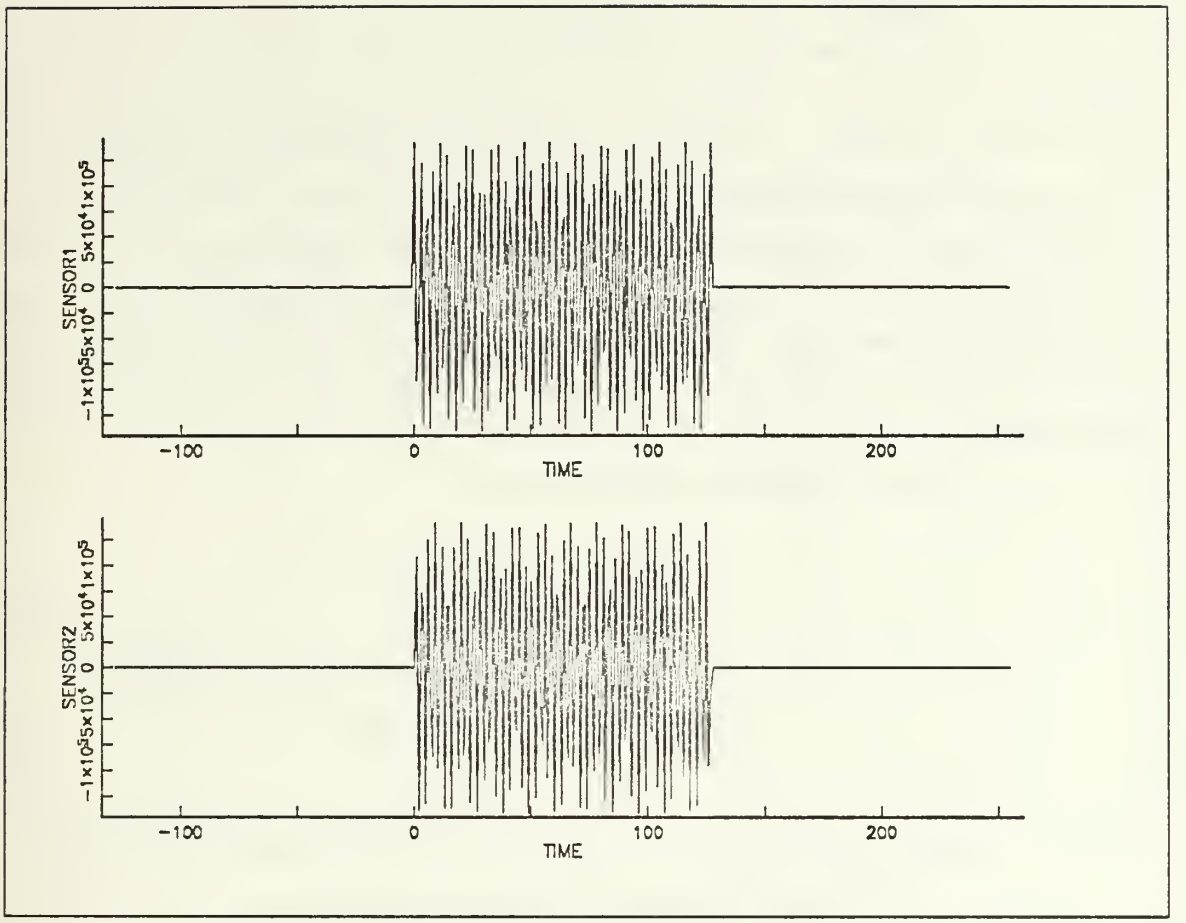

Figure 6. Receiving signals at sensor 1,2 (SNR = 100, no differential delay).

We can estimate the differential time delay and differential Doppler using two different approaches. The first approach uses the cross power spectrum while the second approach uses the coherence.

1. Differential time delay and differential Doppler estimation using the cross power spectrum.

Figure 4 shows the block diagram of the coherence estimator using AR modeling. In this figure, the output of the FFT can be interpreted as the cross power spectrum. Using this cross power spectrum, we can estimate the differential time delay and the differential Doppler. But when we use the highest peak of the cross power spectrum, the peak is somtimes not detected at the proper time delay nor at the proper Doppler shift. We will show a special case in which the peak of the power spectrum is not detected at the proper time delay nor at the proper Doppler values.

a. Special case.

For our test case the data duration is six seconds, two seconds of the noisy signal and four seconds of noise only. Linear transformation of this data leads to one of three types of outputs. The first type represents full information, while the second type represents partial information. The third type represents a noise only condition. Generally when two signals are lined up in time, the AR model should give the highest output power. But this is wrong in some cases. In the full information case, the magnitude of transformed signal is high and constant. For some reduced information case, even though the magnitude of the transformed signal decreases, it still may have large magnitude of spectral components. This phenomenon can be explained as follows.

Define two functions depending on k

$$ {}_kA_y = \sum_{n=0}^{N-1-k} \cos(2\pi f_2 \frac{n}{f_s}) e^{j2\pi f \frac{n}{N}} \tag{4.25} $$

$$ {}_kB_y = \sum_{n=0}^{N-1-k} \sin(2\pi f_2 \frac{n}{f_s}) e^{j2\pi f \frac{n}{N}} \tag{4.26} $$

where $f_2 = f + \alpha_2$.

k denotes the number of lost data at BPF2 input vector.

We assume that we know when the BPF1 signal starts. If this information is not available, we need to examine the signal at the output of BPF1, to obtain a candidate time frame. Consider the input of AR model as shown in Figure 7.

The input data size to the AR model is the number of linearly transformed data points during a given period (i.e., input size is $N = f_s$). The BPF1 output represents full information (i.e., each output element is produced from the signal information points). Therefore, as shown in the previous section, the dominant output sequence is

$$ k_1 e^{j0},\ k_1 e^{j_x \Phi_1},\ k_1 e^{j_x \Phi_2},\ k_1 e^{j_x \Phi_3}, \ldots\ldots,\ k_1 e^{j_x \Phi_{N-1}} \tag{4.27} $$

where

$$ k_1 = \frac{1}{2} \frac{1 - e^{j2\pi\alpha_1}}{1 - e^{j2\pi\alpha_1 \frac{1}{N}}} \tag{4.28} $$

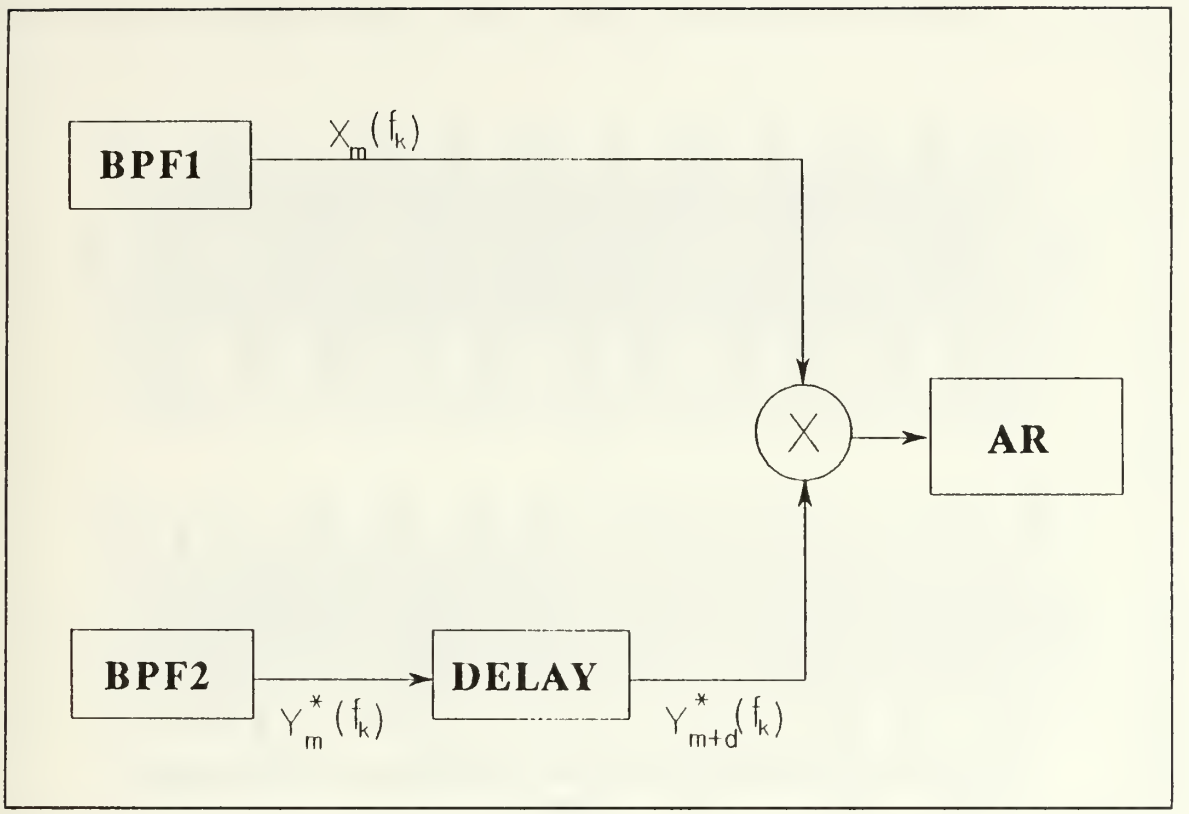

Figure 7. AR model and its driver source.

(1) 0 second delay case. We already defined the input vector $X_m(n)$ and $Y_m(n)$ to BPF1 and BPF2. This vector can be interpreted as a time snap shot. Each output of the BPF requires N input points. If we require N output points from the BPF, even using maximum overlap in the processing, we require at least $2N-1$ points at the BPF input. Figure 8 shows the $2N$ input to BPF1 and BPF2. Both $2N$ input points contain the signal and some noise. As shown in Figure 8, in the zero delay case, the two input vectors $X_m(n)$ and $Y_m(n)$ ($0 \leq m \leq N$) provide full information. So the linearly transformed output $X_m(f_k)$ and $Y_m^*(f_k)$ also represent full information.
The dominant part of the BPF2 output is the sequence

$$k_2 e^{-j0},\ k_2 e^{-j_y\phi_1},\ k_2 e^{-j_y\phi_2},\ k_2 e^{-j_y\phi_3}, \ldots\ldots,\ k_2 e^{-j_y\phi_{N-1}} \tag{4.29}$$

where

$$k_2 = \frac{1}{2}\,\frac{1 - e^{-j2\pi\alpha_2}}{1 - e^{-j2\pi\alpha_2\frac{1}{N}}} \tag{4.30}$$

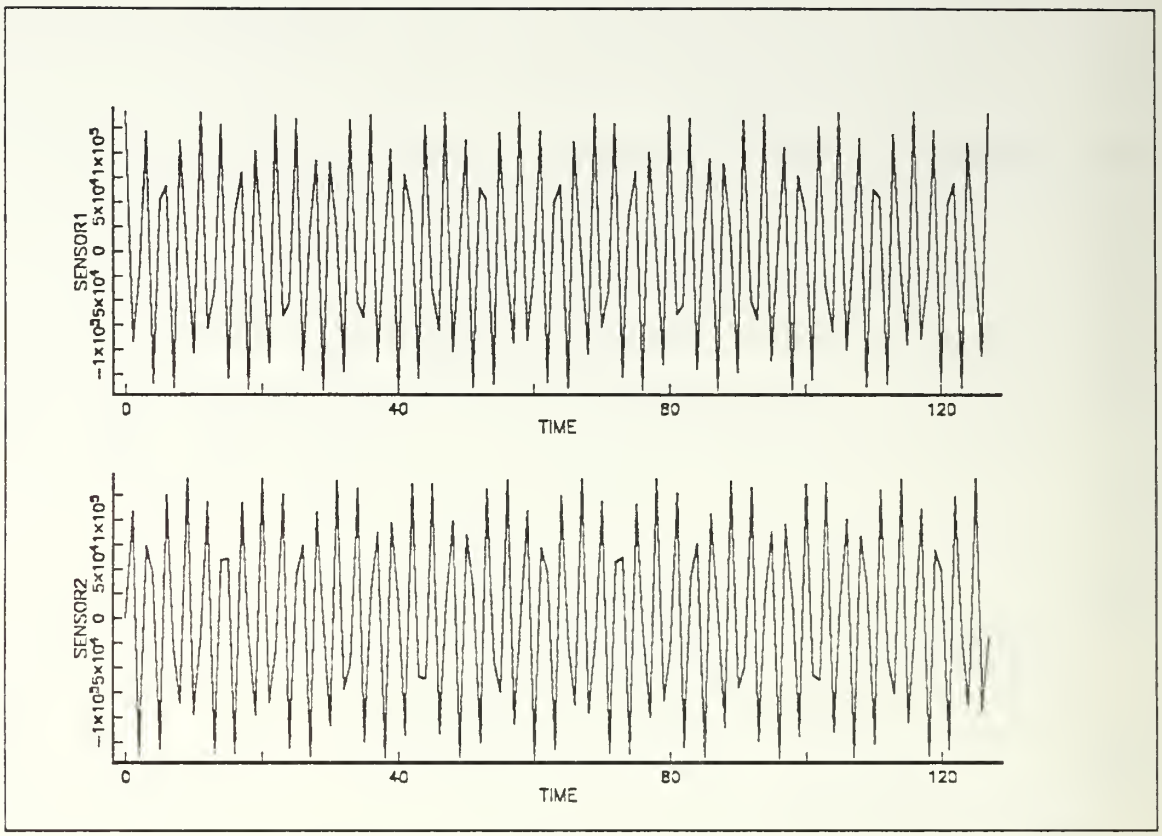

Figure 8. Two signals, 0 second delay at the sensors.

The dominant input sequence to the AR model is given by

$$\{k_1 k_2 e^{j(x\phi_k - y\phi_k)} \quad , k = 0,1,2,\ldots,N-1 \ \} \tag{4.31}$$

where the magnitude is $k_1 k_2$

(2) -1 second delay case. Figure 9 shows the $2N$ input to BPF1 and BPF2. The $2N$ input data points of BPF1 contain the signal and noise. During the first N points the input to BPF2 contains the signal plus noise while during the second N points only noise is present. The input vector $X_m(n)$ of BPF1 contains full information, but input vector $Y_m(n)$ of BPF2 does not contain full information. In the $Y_0(n)$ case, every element of $Y_0(n)$ is full information. But when k is not equal to zero then $Y_k(n)$ consists of $N-k$ information elements and k noise elements.

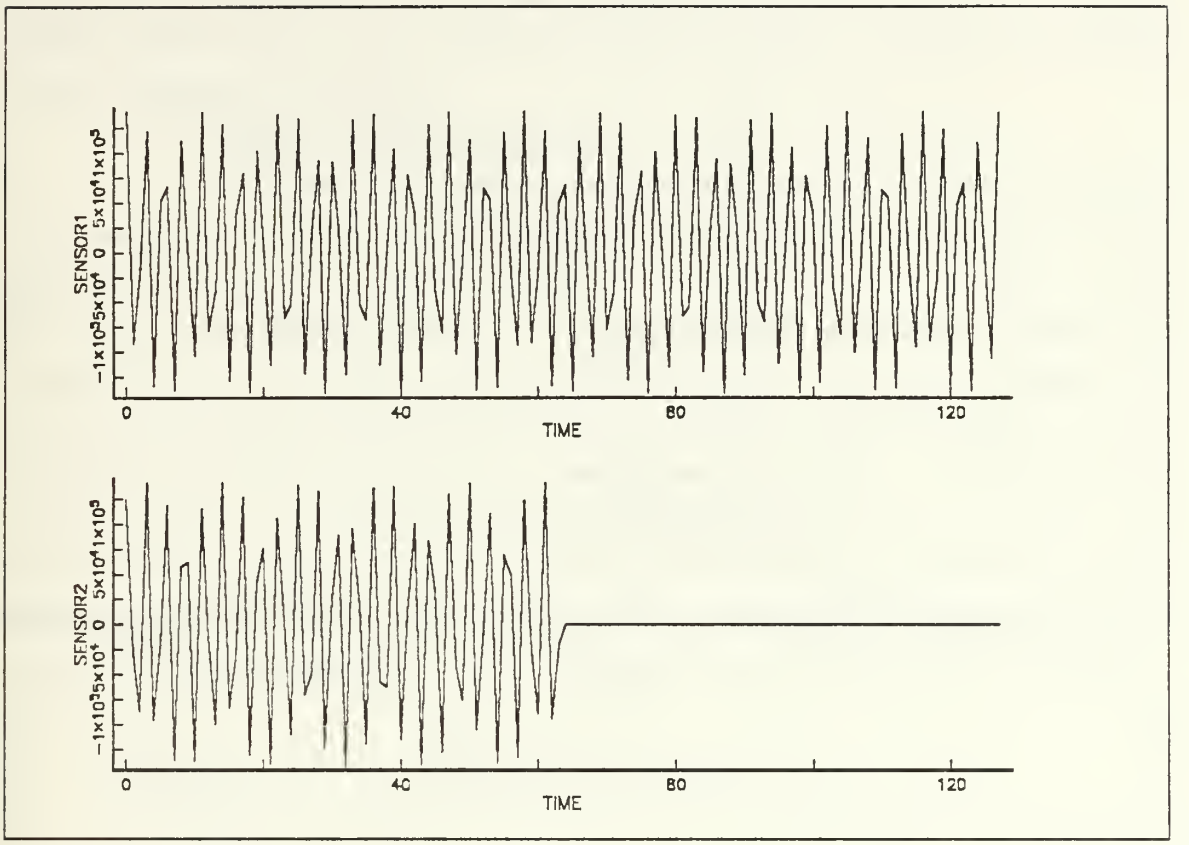

Figure 9. Two signals, -1 second delay at the sensors.

The output from BPF2 represents partial information while the output from BPF1 represents full information. The BPF2 output sequence is given by

$$\{ \frac{{}_kA_y + j_kB_y}{2} e^{j_y\phi_k} + \frac{{}_kA_y - j_kB_y}{2} e^{-j_y\phi_k}, \quad k = 0,1,....N-1 \ \} \tag{4.32}$$

The derivation of Eq. (4.32) is given in Appendix B.

Assuming that $\frac{{}_kA_y - jB_y}{2} e^{-j_y\phi_k}$ is the dominant term, which is a valid assumption for our test case with $f_s = 64Hz$, $f = 23Hz$, $\alpha_1 = 0.4Hz$ and $\alpha_2 = 0.001Hz$, then the input sequence to the AR model is dominated by

$$\{k_1 \frac{{}_kA_y - j_kB_y}{2} e^{j({}_x\phi_k - {}_y\phi_k)}, \quad k = 0,1,2,.....,N-1\} \tag{4.33}$$

with k_1 is defined earlier (Eq. (4.28)).

The magnitude of dominant input sequence is obtained by examining

$$
\begin{aligned}
{}_kA_y - j_kB_y &= \sum_{n=0}^{N-1-k} [\cos\{2\pi(f+\alpha_2)\} - j\sin\{2\pi(f+\alpha_2)\}] e^{j2\pi f\frac{n}{N}} \\
&= \sum_{n=0}^{N-1-k} e^{-j2\pi(f+\alpha_2)\frac{n}{f_s}} e^{j2\pi f\frac{n}{N}} \\
&= \sum_{n=0}^{N-1-k} e^{-j2\pi\alpha_2\frac{n}{N}} \\
&= \frac{1-e^{-j2\pi\frac{N-k}{N}\alpha_2}}{1-e^{-j2\pi\alpha_2\frac{1}{N}}}
\end{aligned}
\tag{4.34}
$$

When we compare the magnitude of the sequence in Eq. (4.27) and the magnitude of the dominant term in Eq. (4.32), the magnitude of $|k_2| > |{}_kA_y - j_kB_y|/2$ where

$$k_2 = \frac{1}{2}\frac{1-e^{-j2\pi\alpha_2}}{1-e^{-j2\pi\alpha_2\frac{1}{N}}}$$

and

$$
\begin{aligned}
\frac{{}_kA_y - j_kB_y}{2} &= \frac{1}{2}\frac{1-e^{-j2\pi\frac{N-k}{N}\alpha_2}}{1-e^{-j2\pi\alpha_2\frac{1}{N}}} \\
&= \frac{1}{2}\frac{e^{j0}}{1-e^{-j2\pi\alpha_2\frac{1}{N}}} - \frac{1}{2}\frac{e^{-j2\pi\alpha_2}}{1-e^{-j2\pi\alpha_2\frac{1}{N}}} e^{j2\pi\alpha_2\frac{k}{N}}
\end{aligned}
\tag{4.35}
$$

We note that Eq. (4.35) has two frequencies (i.e., 0 and $\frac{\alpha_2}{N}2\pi$).

The magnitudes of both terms are $\frac{1}{|1-e^{-j2\pi\alpha_2\frac{1}{N}}|}$

Now, note that as long as $|\alpha_2| < \frac{1}{6}$, then the magnitudes of both terms in Eq. (4.35) are larger than $|k_2|$

Due to the symmetry a positive delay results in a similar reponse to a negative delay, and hence Eq. (4.35) holds also for delay of +1 second. Equation (4.35) can be interpreted as having large responses at shifted Doppler values at a delay of 1 second and a delay

of −1 second. Proper delay can not be established nor can the Doppler value be established. This cross power spectrum algorithm will not give good information about either Doppler difference or time delay.

b. Modified cross power spectrum.

To detect the signal, input signals to the AR model should be preprocessed as shown Figure 10. If the BPF2 output signal is magnitude normalized, then this normalized signal retains good phase information. When we normalize the BPF2 output, one of two conditions can occur. The first condition (i.e., synchronized) leads to the magnitude normalization of Eq. (4.29) and provides accurate frequency and delay estimation. The second condition occurrs when the signal in channel-2 is not lined up in time with the signal in channel-1 (i.e., during the delay search). Under this condition smaller peaks in the time Doppler plane appear at incorrect values of time delay and Doppler. Above 70 dB *SNR*, the correct peak is the dominant one and allows proper estimation. Information from contour plots can be used at values of *SNR* between 70 and 10 dB (see Appendix D).

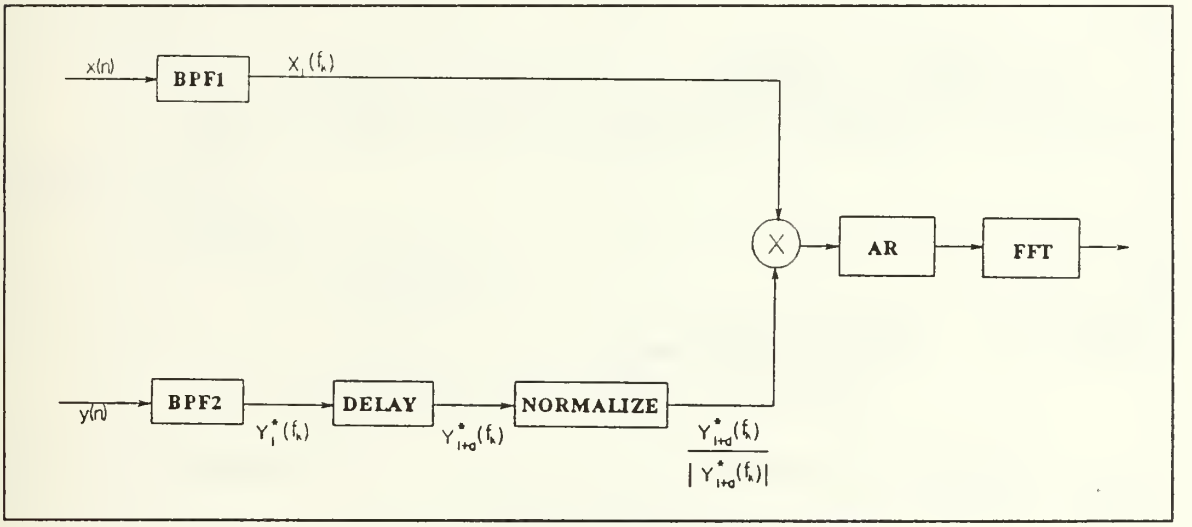

Figure 10. Modified cross power spectrum block diagram.

2. Differential time delay and differential Doppler estimation using the coherence.

Another way detecting the signal is normalization of the AR output with the squared root of the product of $P_{xx}(f)$ (auto spectrum of x) and $P_{yy}(f)$ (auto spectrum of y). This can be done using two additional AR models as shown in Figure 11.

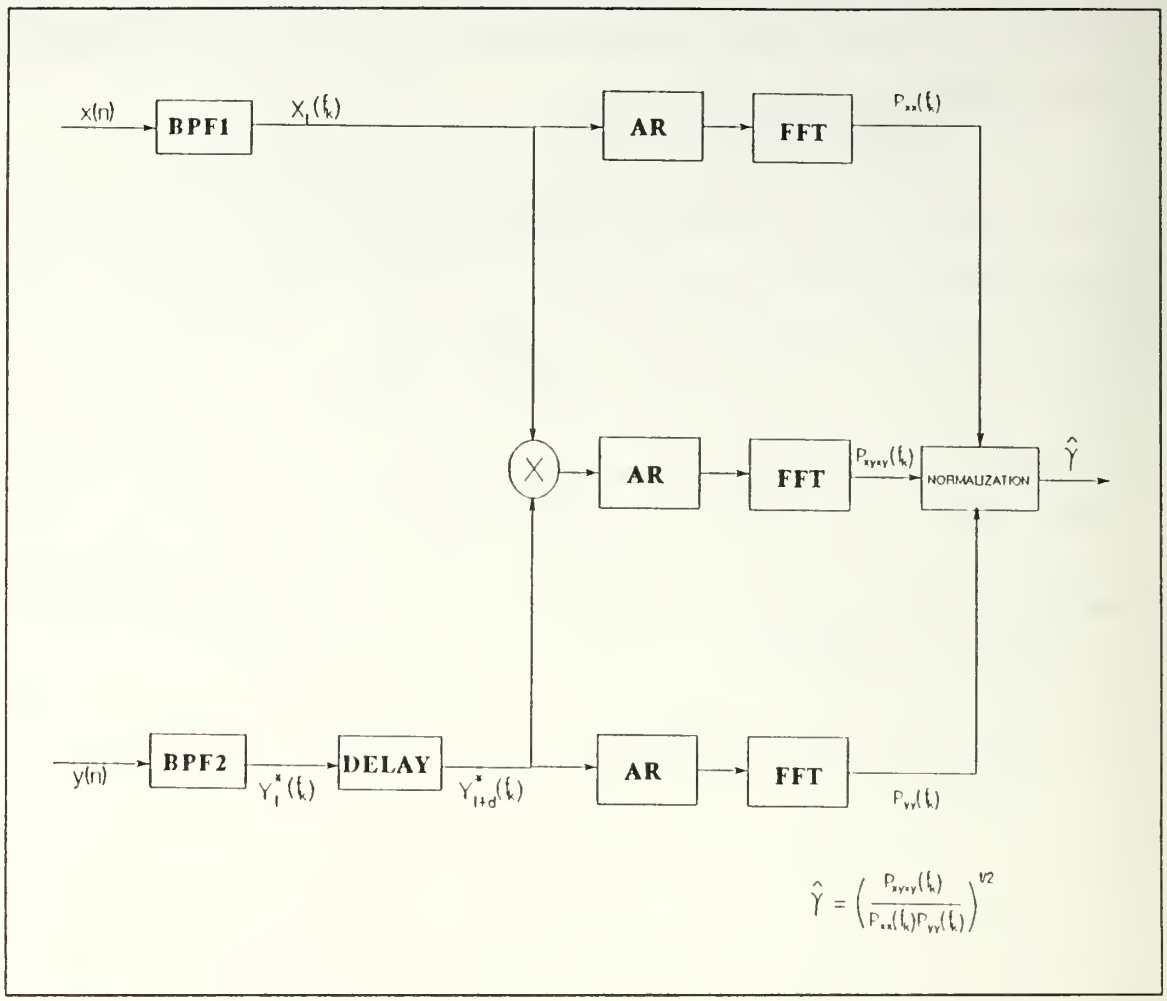

Figure 11. AR model of coherence.

Using the AR approach, ρ_{xx} , ρ_{xyxy} , ρ_{yy} , and AR coefficients can be obtained. Therefore we get

$$P_{xx}(f) = \frac{T\rho_{xx}}{|A_{xx}(f)|^2} \tag{4.36}$$

$$P_{yy}(f) = \frac{T\rho_{yy}}{|A_{yy}(f)|^2} \tag{4.37}$$

$$P_{xyxy}(f) = \frac{T\rho_{xyxy}}{|A_{xyxy}(f)|^2} \tag{4.38}$$

$$P_{yy}(f) = \frac{T\rho_{yy}}{|A_{yy}(f)|^2} \tag{4.37}$$

$$P_{xyxy}(f) = \frac{T\rho_{xyxy}}{|A_{xyxy}(f)|^2} \tag{4.38}$$

where $P_{xx}(f)$ is the power spectrum of X_l,

$P_{yy}(f)$ is the power spectrum of Y_{l-d},

$P_{xyxy}(f)$ is the power spectrum of $X_l Y_{l-d}$,

$A_{xx}(f)$ is the AR power spectral density of X_l,

$A_{yy}(f)$ is the AR power spectral density of Y_{l-d}.

$A_{xyxy}(f)$ is the AR power spectral density of $X_l Y_{l-d}$ cross term.

ρ_{xx} is the driving noise variance of X_l.

ρ_{yy} is the driving noise variance of Y_{l-d},

ρ_{xyxy} is the driving noise variance of $X_l Y_{l-d}$, and

$T = \frac{1}{f_s}$.

We obtain a coherence estimate from the three power spectra, by assuming that $|P_{xy}(f)|^2 \approx |P_{xyxy}(f)|$.

$$\begin{aligned} \hat{\gamma}^2 &= \frac{|P_{xy}(f)|^2}{|P_{xx}(f)P_{yy}(f)|} \\ &\approx \frac{|P_{xyxy}(f)|}{|P_{xx}(f)P_{yy}(f)|} = \frac{1}{T}\frac{\rho_{xyxy}}{\rho_{xx}\rho_{yy}}\frac{|A_{xx}(f)A_{yy}(f)|^2}{|A_{xyxy}(f)|} \end{aligned} \tag{4.39}$$

$$\hat{\gamma} = \sqrt{\frac{1}{T}\frac{\rho_{xyxy}}{\rho_{xx}\rho_{yy}}\frac{|A_{xx}(f)A_{yy}(f)|^2}{|A_{xyxy}(f)|}} \tag{4.40}$$

This approach provides good information about the differential time delay and differential Doppler down to *SNRs* of 20 dB.

V. RESULTS

This chapter presents graphical results obtained by applying the analytical results of the previous chapters to specific examples and carrying out the required computations on the computer.

A. AR MODEL

This section provides the AR model performance tests. As we discussed in chapter III, Burg's algorithm is applied to this AR model. The FPE criterion is use to find the AR model order. Figure 12 is the power spectrum of two test signals $\sin(\omega_1 Tn)$ and $\cos(\omega_1 Tn)$. Figure 13 shows the power spectrum of two other test signals $\sin(\omega_2 Tn)$ and $\cos(\omega_2 Tn)$. Both sets of test signals in these figures have an SNR of 20 dB. The frequency f_1 is 13.45 Hz and f_2 is 23.45 Hz and T is $\frac{1}{64}$ second.

Theoretically, the power spectrum of these signals have two impulse functions at f_i and $-f_i$ ($i = 1, 2$). Figures 12 and 13 show experimental results agreeing with the theoretical results. Figures 12 and 13 indicate which frequency has high power but they provide no phase information.

The AR model is sensitive to frequency but is not sensitive to the phase. For any selection of frequency f which satisfies the Nyquist theorem, the AR model provides good frequency estimation provided the SNR is sufficient large.

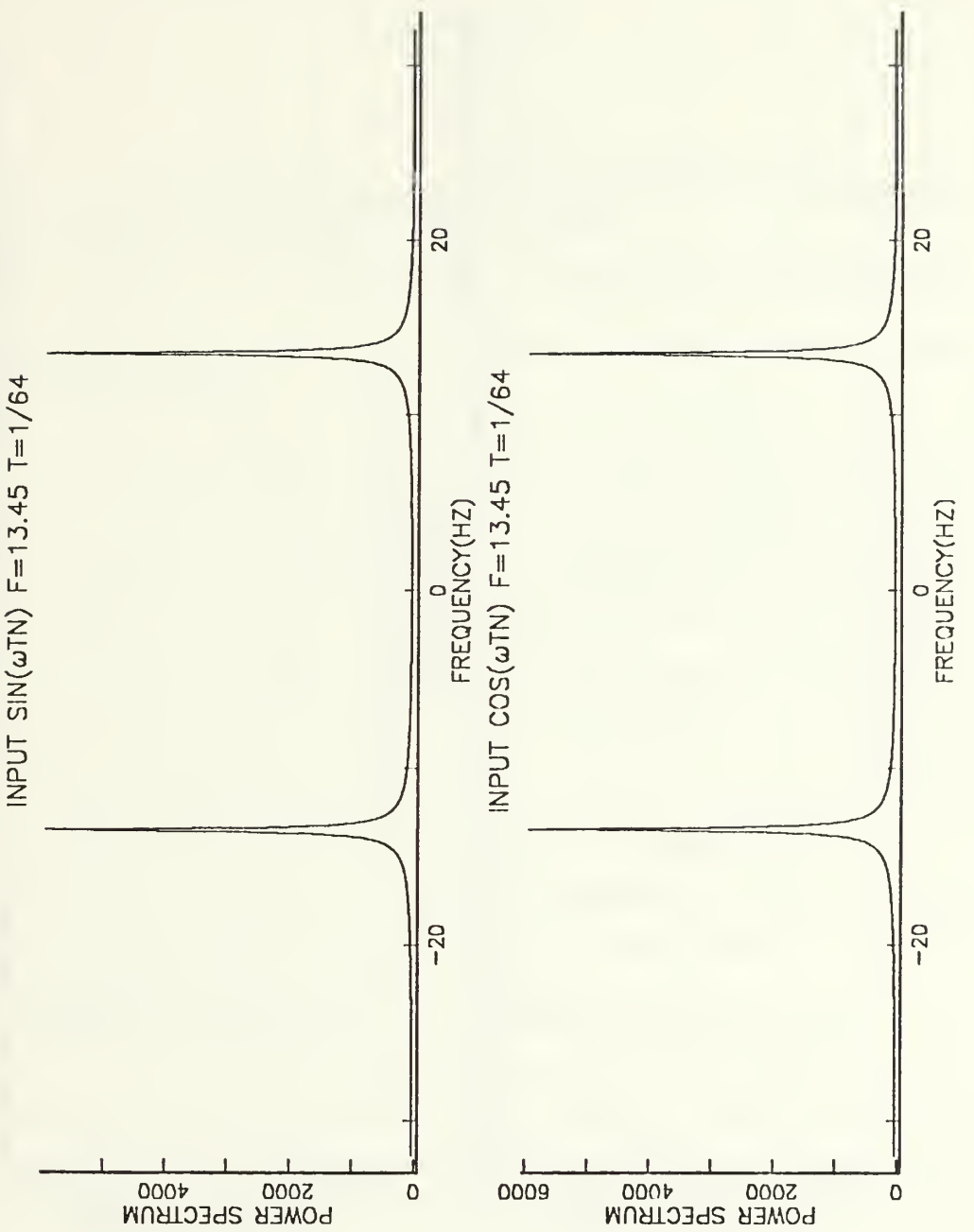

Figure 12. AR model performance test 1 (SNR = 20 dB).

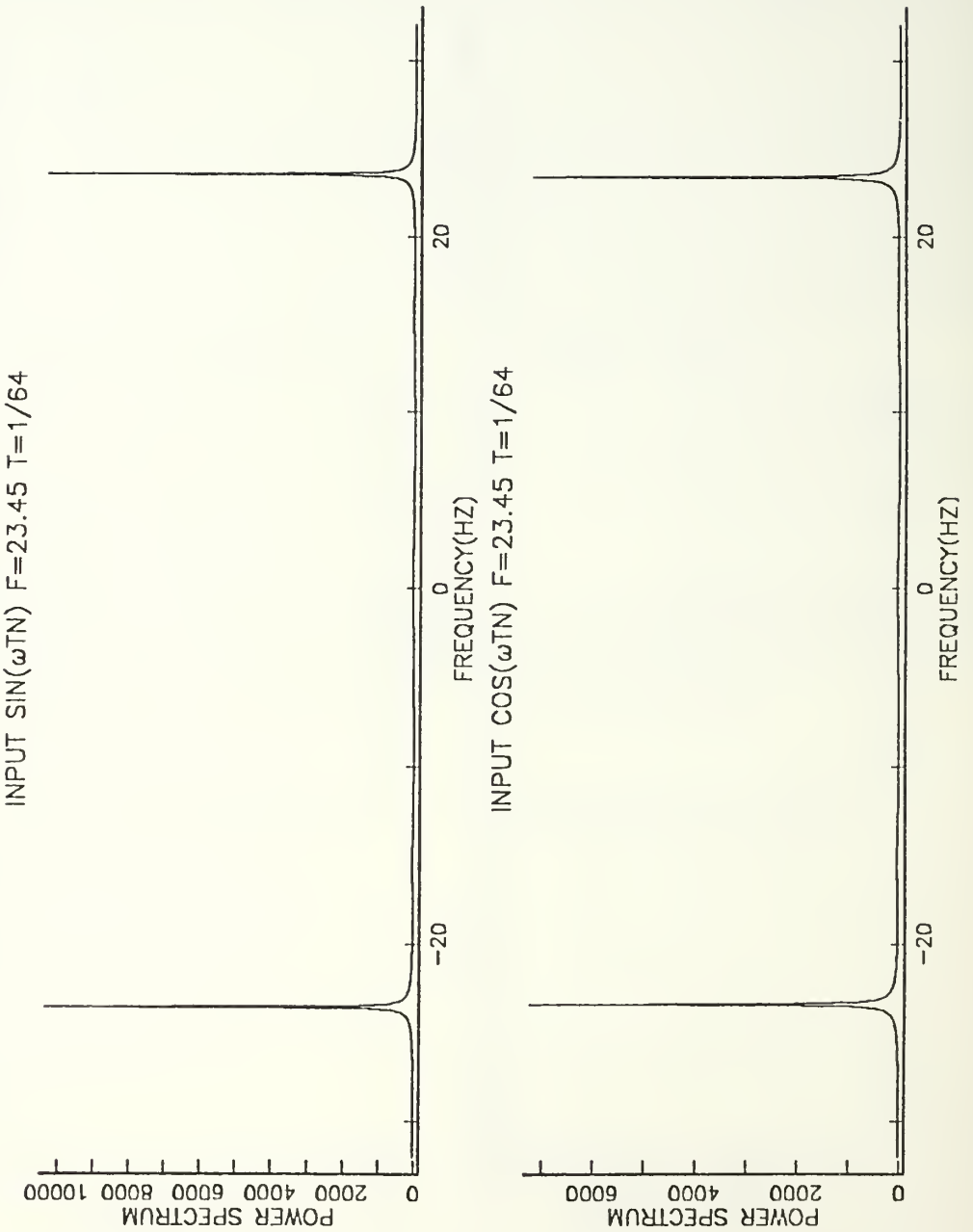

Figure 13. AR model performance test 2 (SNR = 20 dB).

B. DOPPLER ESTIMATION

In section A of chapter IV, we discussed five items.

1. BPF1 and BPF2 outputs have two sinusoidal components each.
2. For BPF1, the dominant frequency is designated as ω_x and for BPF2, the dominant frequency is designated as ω_y where

$$\omega_x = \frac{2\pi(f+\alpha_1)}{f_s}$$

$$\omega_y = \frac{-2\pi(f+\alpha_2)}{f_s}$$

3. The power spectrum of cross term of BPF1 and BPF2 has four components which can be detected in very high *SNR* (i.e. *SNR* = 100 dB).
4. When the *SNR* is low (i.e. *SNR* = 20 dB), only one dominant frequency component is detected.
5. The dominant frequency of the power spectrum $P_{xyxy}(f)$ corresponds to the differential Doppler frequency.

Figure 14 shows the power spectrum of the BPF1 output when f is 13 *Hz*, α_1 is 0.45 *Hz*, and f_s is 64 *Hz*. Figure 15 shows the power spectrum of the BPF2 output when f is 13 *Hz*, α_2 is -0.45 *Hz*, and f_s is 64 *Hz*. As expected both figures show two spectral lines when the *SNR* is 100 dB. The dominant frequencies are located at f equal 13.45 *Hz* for BPF1, and at f equal -12.55 *Hz* for BPF2. Figure 16 shows the power spectrum of the cross term of BPF1 and BPF2. This figure shows four spectral lines at an *SNR* of 100 dB, while only one dominant frequency is detected at an *SNR* of 20 dB. Dominant frequencies are always located between -1 *Hz* and 1 *Hz*. Figure 17 shows a subplot of Figure 16 for frequency between -1 *Hz* and 1 *Hz*. This figure shows the dominant frequency $f = .9$ *Hz* and the smaller spectral component at $f \approx -.9$ *Hz* for an *SNR* of 100 dB. Only one dominant frequency can be detected at f equal 0.9 *Hz* for an *SNR* of 20 dB. Doppler difference from the dominant spectral component corresponds to the true Doppler difference $\alpha_1 - \alpha_2$ which is 0.9 *Hz*

Figure 18 shows a comparison for *SNR* of 20 dB and 10 dB. Figure 19 is subplot of Figure 18 for -1 *Hz* $< f <$ 1 *Hz*. From this figure, when *SNR* = 10 dB, the dominant frequency is shifted a little bit from the true Doppler difference. We have shown the five issues as we have discussed in the previous chapter. In summary, for *SNR*s greater than or equal to 20 dB, the power spectrum of the cross term provides good Doppler estimation.

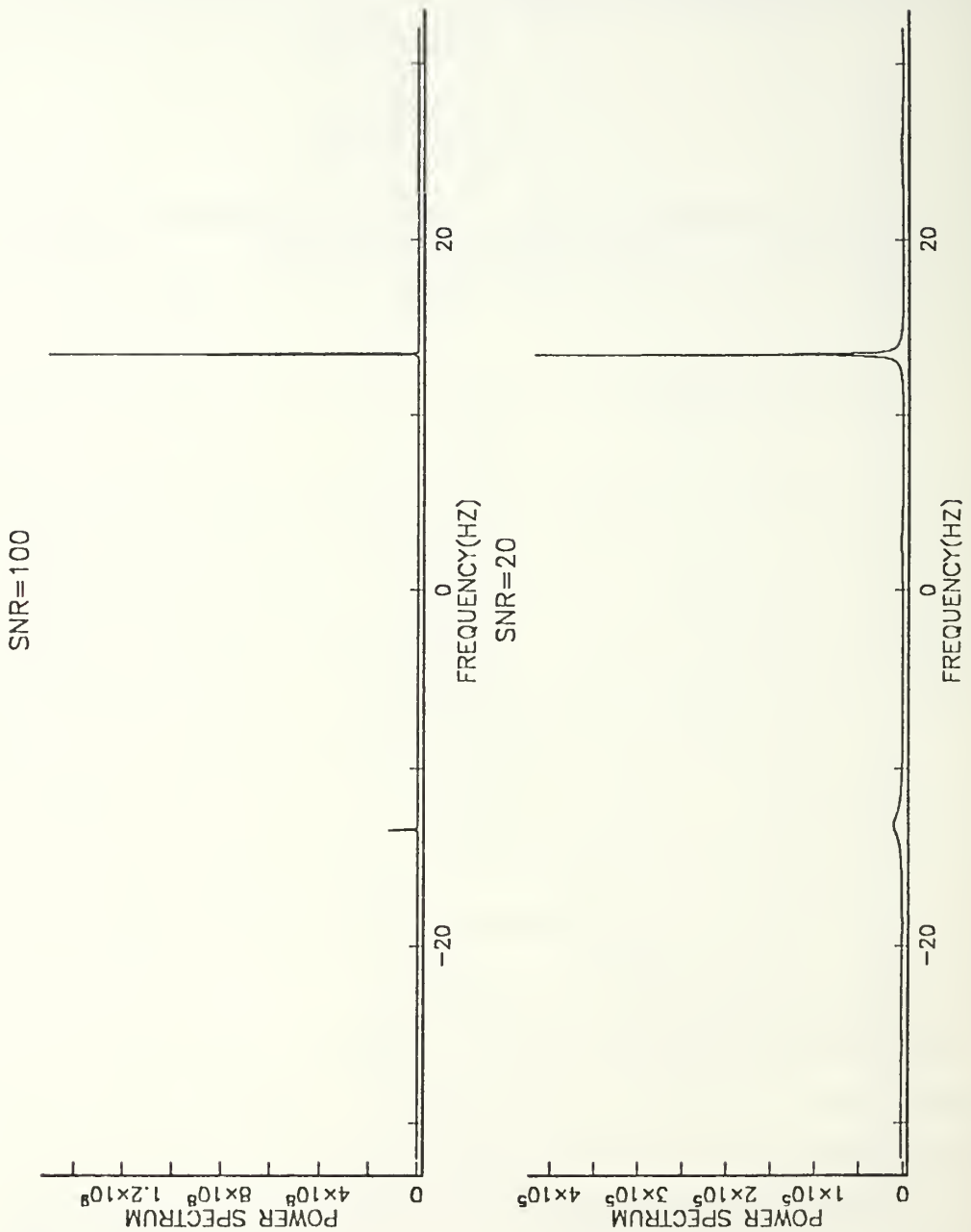

Figure 14. Power spectrum of the BPF1 output.

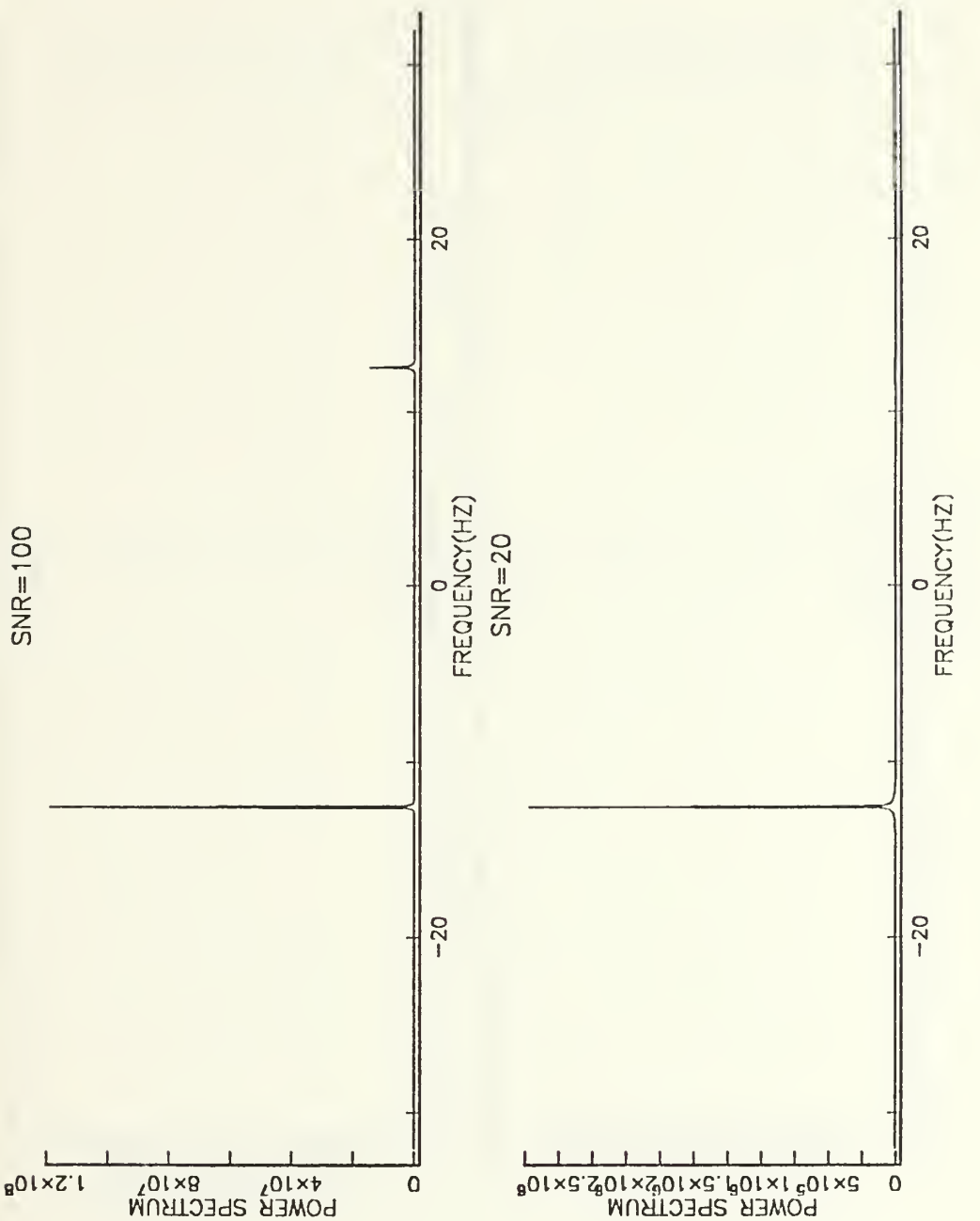

Figure 15. Power spectrum of the BPF2 output.

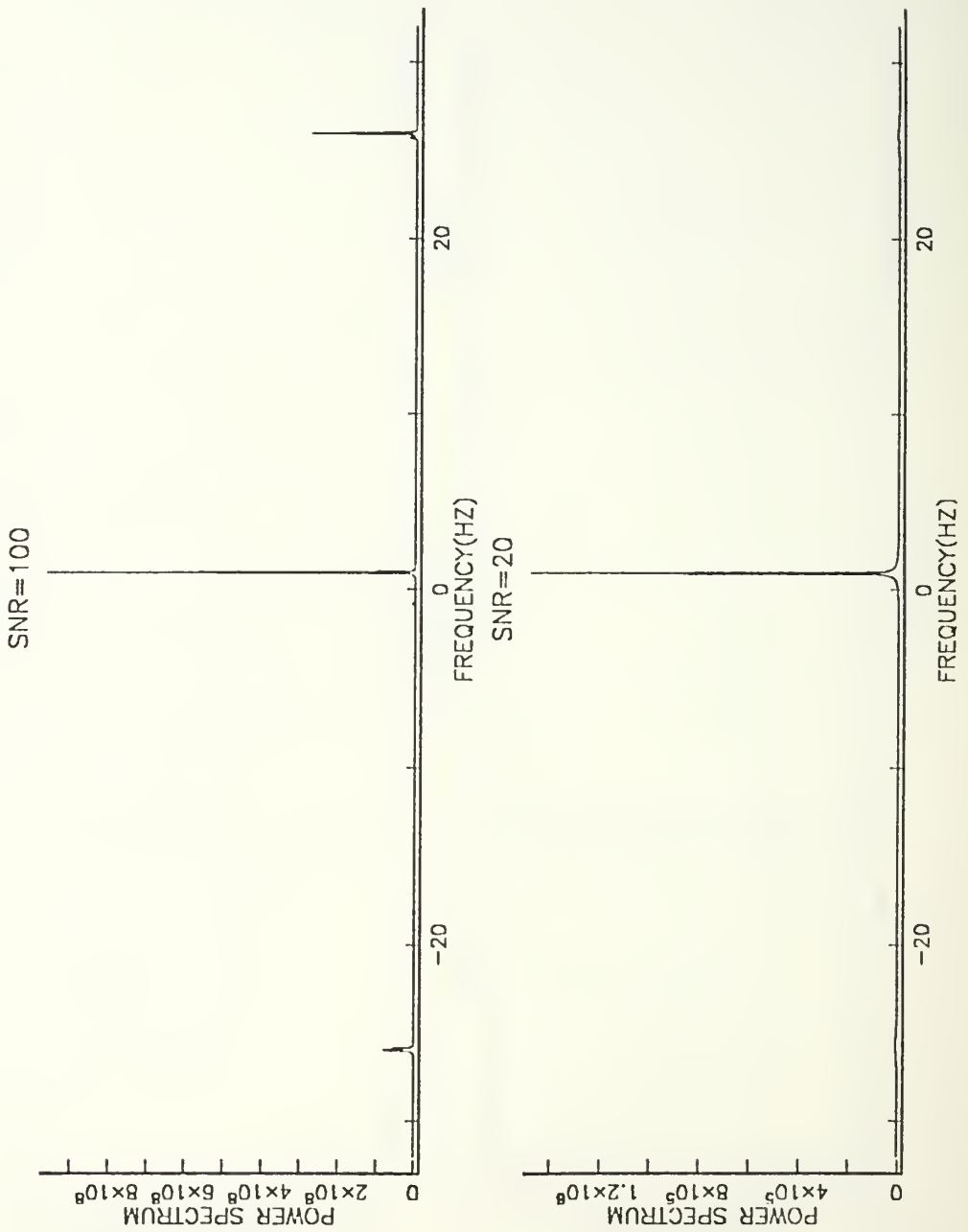

Figure 16. Cross power spectrum of BPF1 and BPF2.

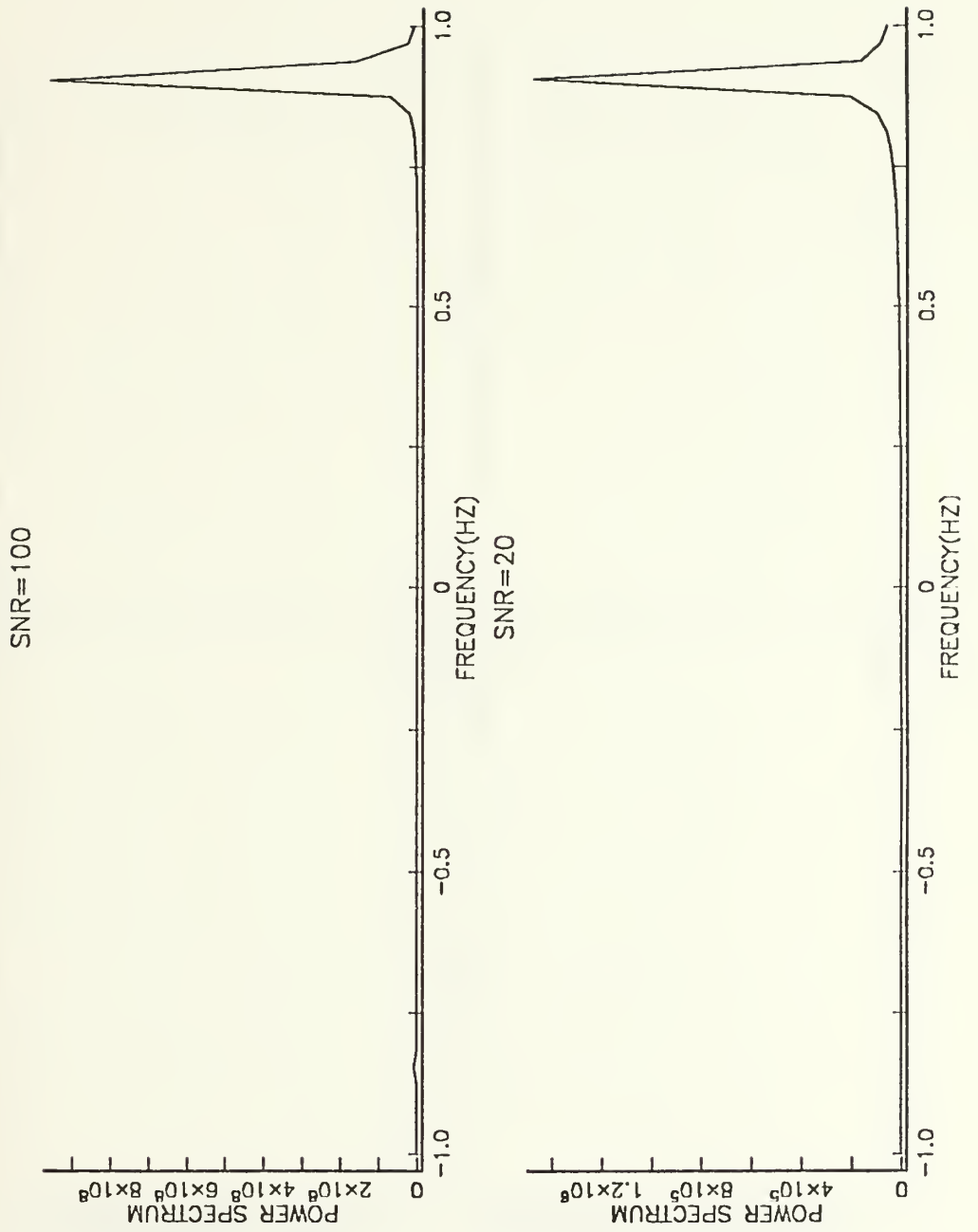

Figure 17. Subplot of the cross power spectrum.

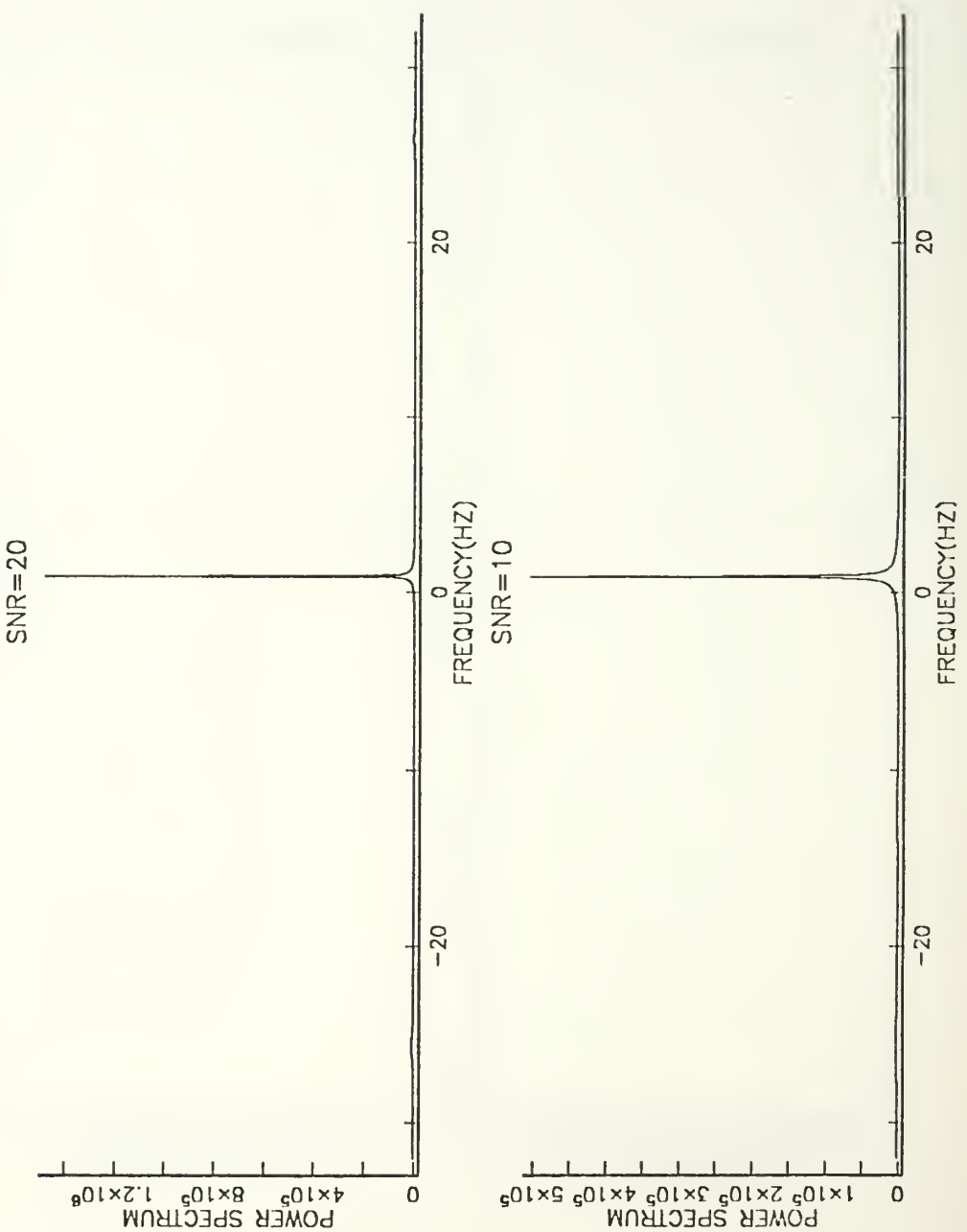

Figure 18. . Comparison the Doppler estimation at two different SNR's.

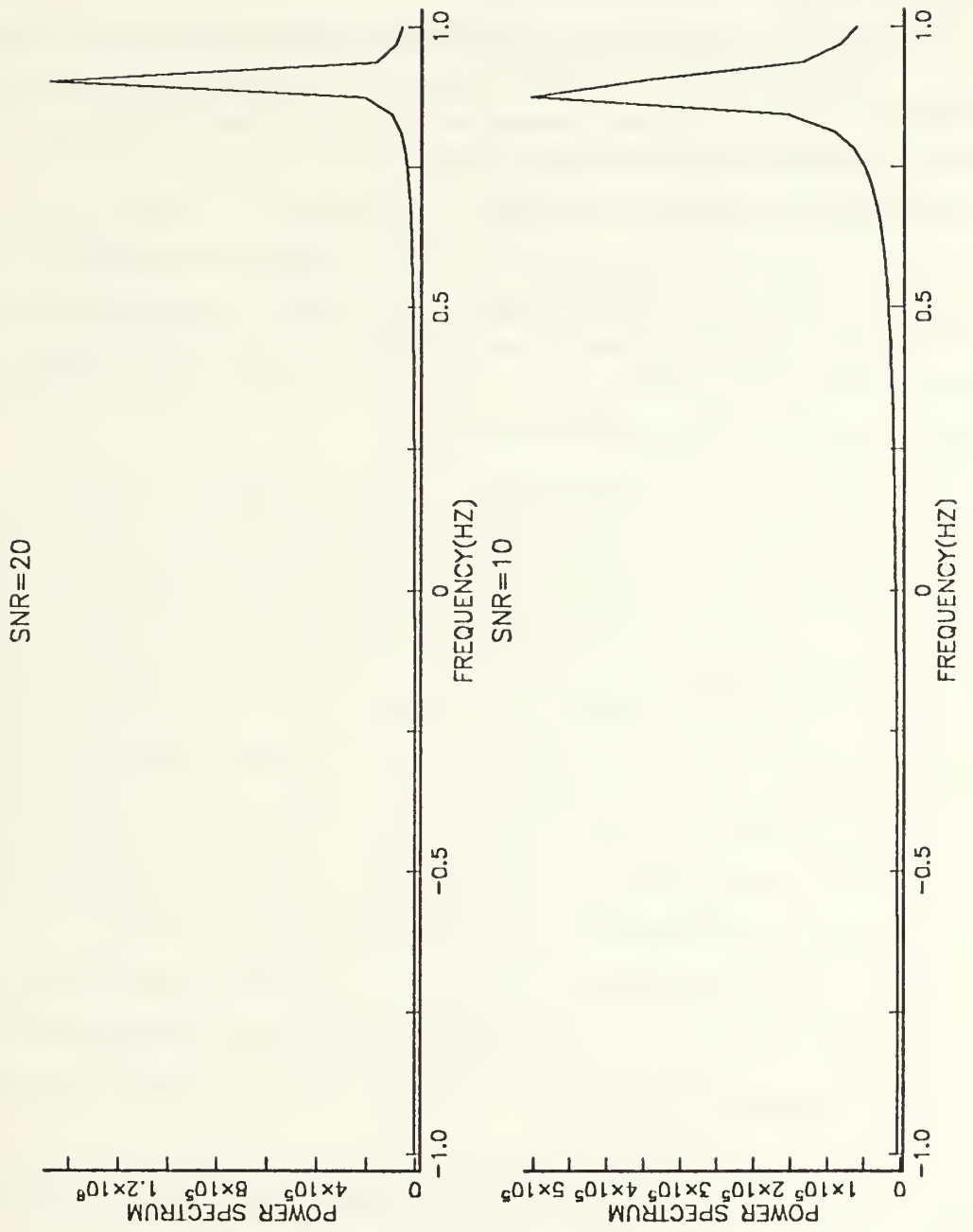

Figure 19. Subplot of the Doppler estimation.

C. DIFFERENTIAL TIME DELAY ESTIMATION

In Chapter IV section B.1.a we discussed the problems associated with using unnormalized quantities and suggested two possible approaches to solve the problem. In this section we will show that for the case of $|\alpha_2| < \frac{1}{6}$ Hz , the power spectrum of a nondelayed signal is smaller than the power spectrum of a signal delayed by one second (i.e., see discussion following Eq. (4.35)).

To estimate the time delay, we suggest the two different approaches. The first approach is the normalization of the BPF2 output. The second approach normalizes the cross term $P_{xyxy}(f)$ with the power spectrum of BPF1 $P_{xx}(f)$ and the power spectrum of BPF2 $P_{yy}(f)$ through the equation defined by Eq. (4.40).

The following figures are simulations for $f = 23$ Hz, $\alpha_1 = 0.23$ Hz, $\alpha_2 = -0.02$ Hz, $T = \frac{1}{64}$ second, delay of 0 and an SNR of 100 dB. Figure 20 shows the surface plot of $P_{xyxy}(f)$ while Figure 21 shows the contour plot of $P_{xyxy}(f)$. Figure 22 shows the cross section plot of Figure 20 with respect to the DELAY axis. This figure shows that the power spectrum has a spurious peak at delay of one second. At the proper delay (i.e., 0 second delay) $P_{xyxy}(f)$ has a relatively small value.

$P_{xyxy}(f)$ is affected by two terms, namely ρ_ω and $\frac{1}{|A_{xyxy}(f)|^2}$. Figure 23 shows the maximum power spectrum term of the transfer function and Figure 24 shows the driving noise variance. At a delay of $\pm$ 1 second and of 0 second the transfer function shows a peak while the variance of driving noise has small value. In this case, $P_{xyxy}(f)$, the product of the form $\rho_\omega \frac{1}{|A_{xyxy}(f)|^2}$ at 0 second delay has a smaller value than that at any other delay location. Using the scheme shown in Figure 10, we normalize the BPF2 output and present the result in Figure 25. We can detect the proper time delay and the proper Doppler difference. Figures 26 and 27 show the corresponding contour plot and the power spectrum of the normalized cross term. For any value of time delay and Doppler difference, this approach provides good information about differential Doppler and differential time delay provided the SNR is greater than or equal to 70 dB.

Results of the second approach (Figure 11) are demonstrated in Figures 28 and 29. This approach also gives good information about differential time delay and differential Doppler. This approach provides good time delay and Doppler estimation for SNR greater than or equal to 20 dB.

If we use the contour shape, we can also estimate the differential time delay and differential Doppler for SNR of less than 20 dB. Differential time delay and differential Doppler estimation can be extended down to $SNRs$ of about 0 dB when using the contour of the coherence surface. Figure 30 is a contour plot of using this second approach

(i.e., coherence approach) for $\alpha_1 = 0.23$ *Hz*, $\alpha_2 = -0.02$ *Hz*, *Delay* = 0, and an *SNR* of 0 dB. The contour shape of the second approach has the shape of the letter X. When the *SNR* is greater or equal to 0 dB, the proper time delay and Doppler difference seems to be located at the cross over point of the X. When the *SNR* is less than 0 dB, the contour shape does not shows an X clearly. In the coherence approach, Figure 31 provides the contour plot, with channel-1 containing only noise and channel-2 containing signal plus noise ($SNR = 0$ dB). Figure 32 shows the contour plot, when channel-2 contains only noise and channel-1 contains signal plus noise ($SNR = 0$ dB). Figure 33 shows the contour plot, when both channels contain noise only. The above three figures do not show an X clearly, hence do not allow signal detection for estimation of any parameter. Using this information, we can distinguish the signal combinations from the noise combinations.

When we estimate the time delay and Doppler difference using the contour plot, the modified cross power spectral approach has some problems as discussed in Appendix D.

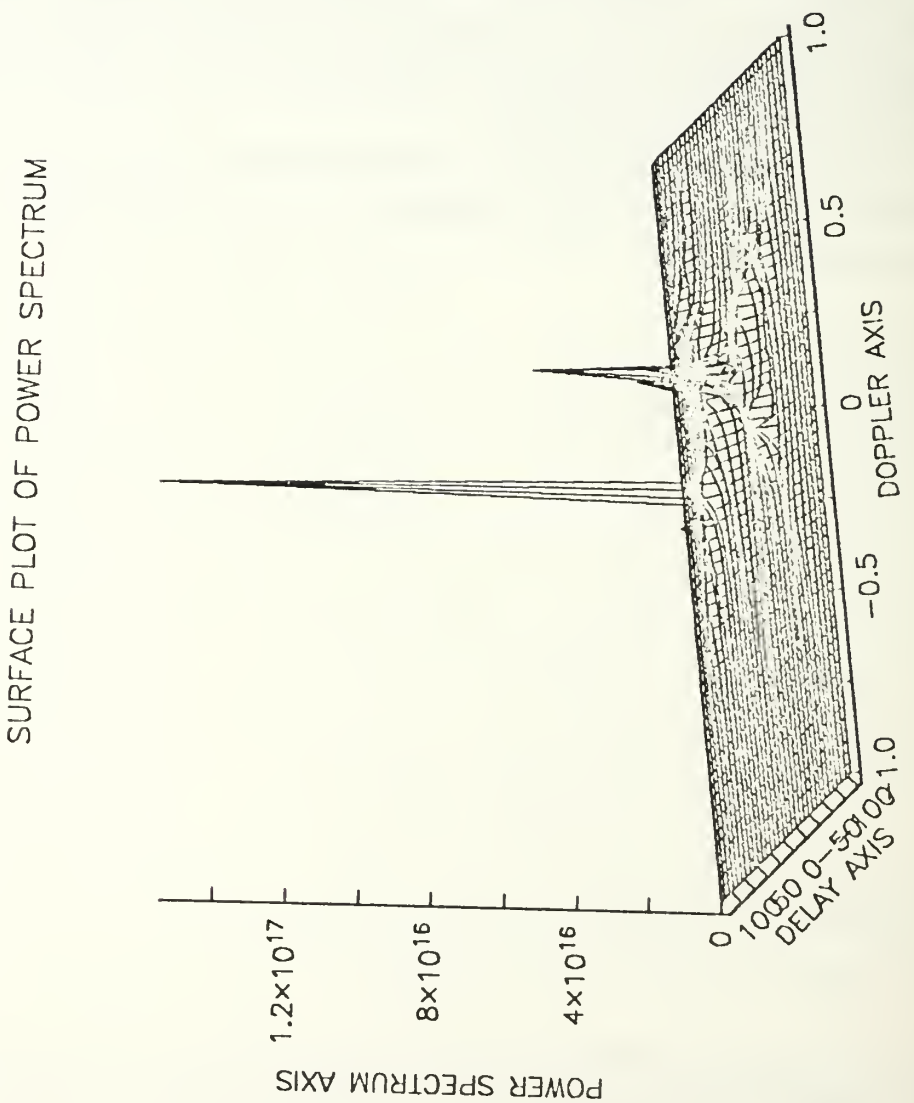

Figure 20. Surface plot of the power spectrum.

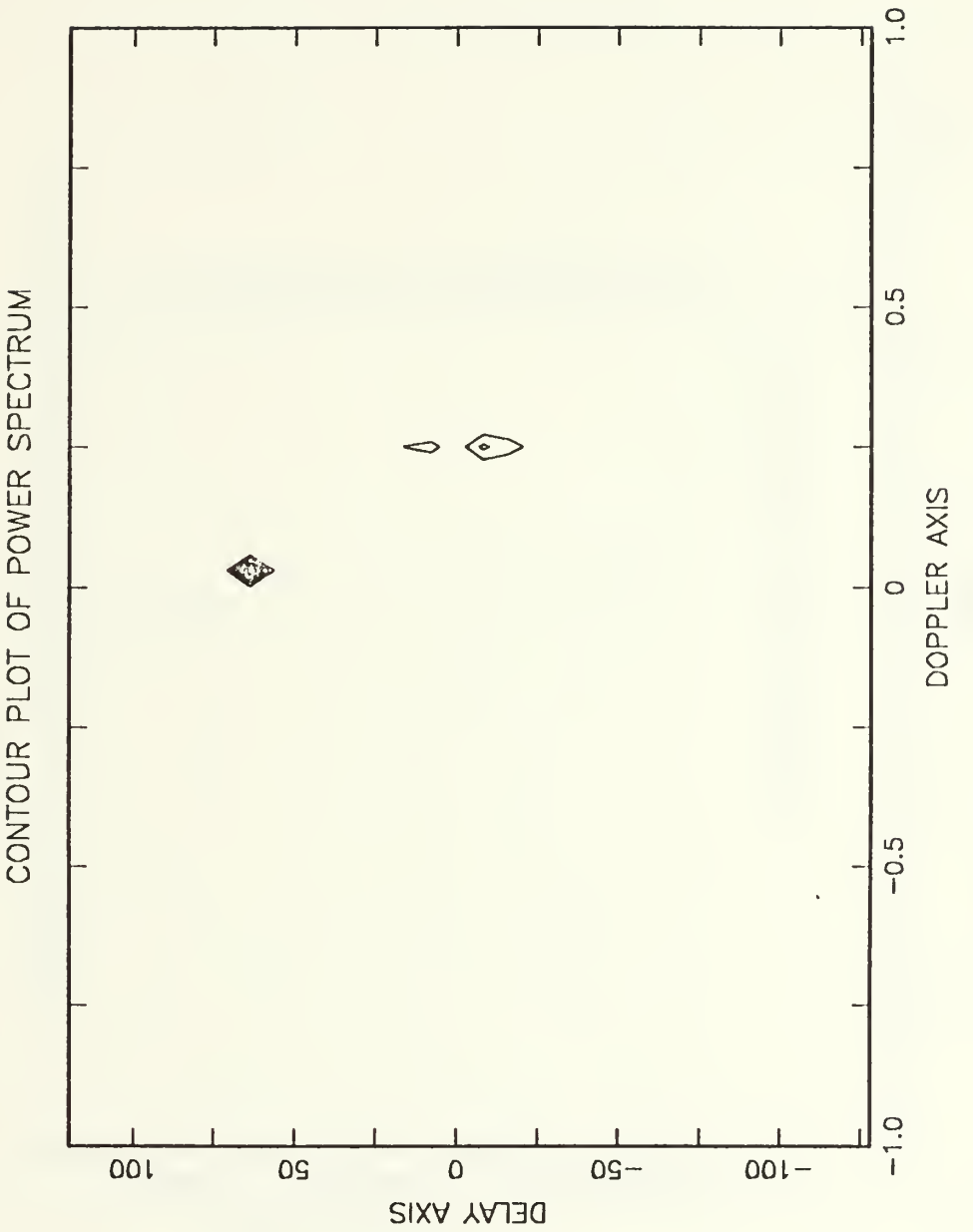

Figure 21. Contour plot of the power spectrum.

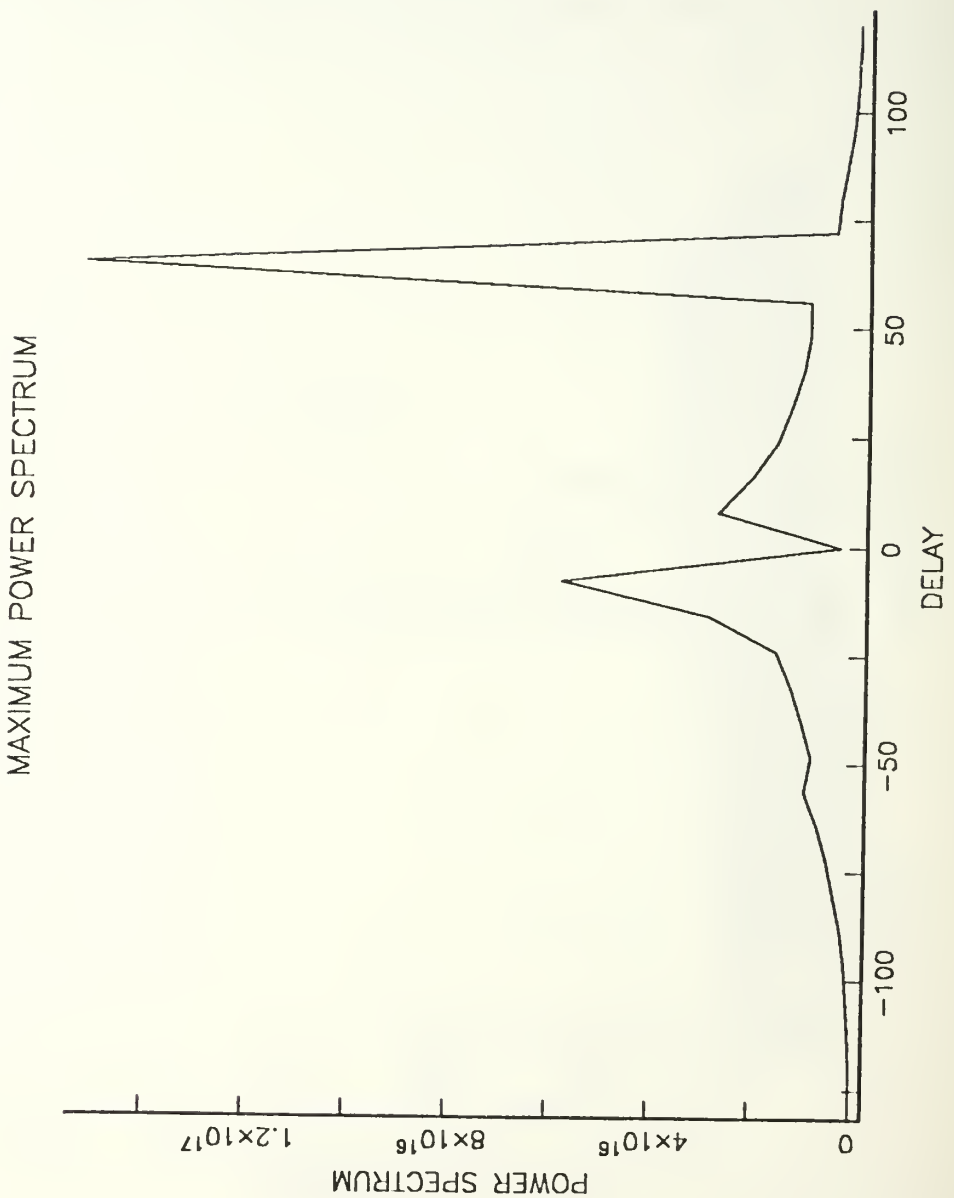

Figure 22. Maximum power spectrum.

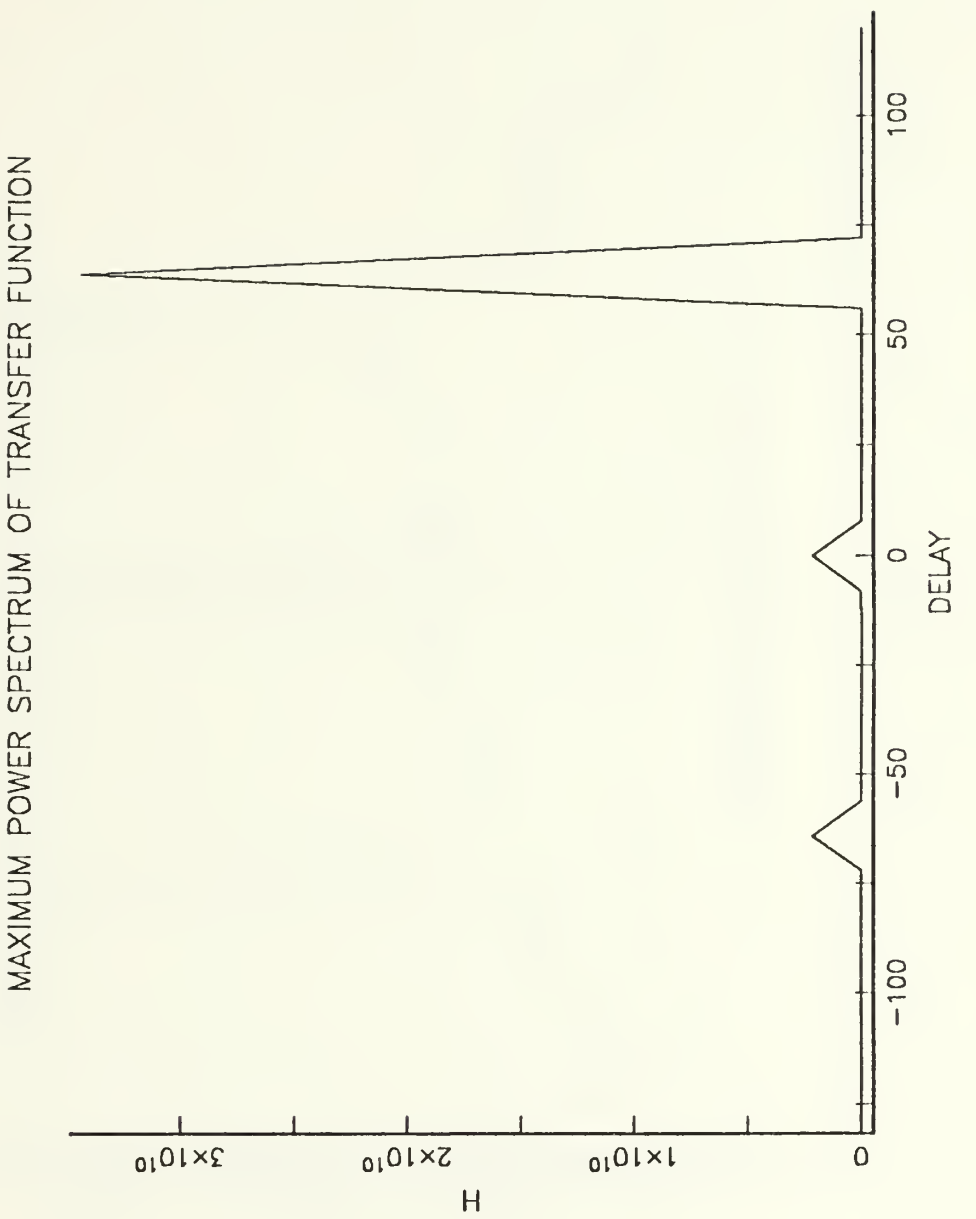

Figure 23. Maximum power spectrum of the transfer function.

Figure 24. Variance of the driving noise.

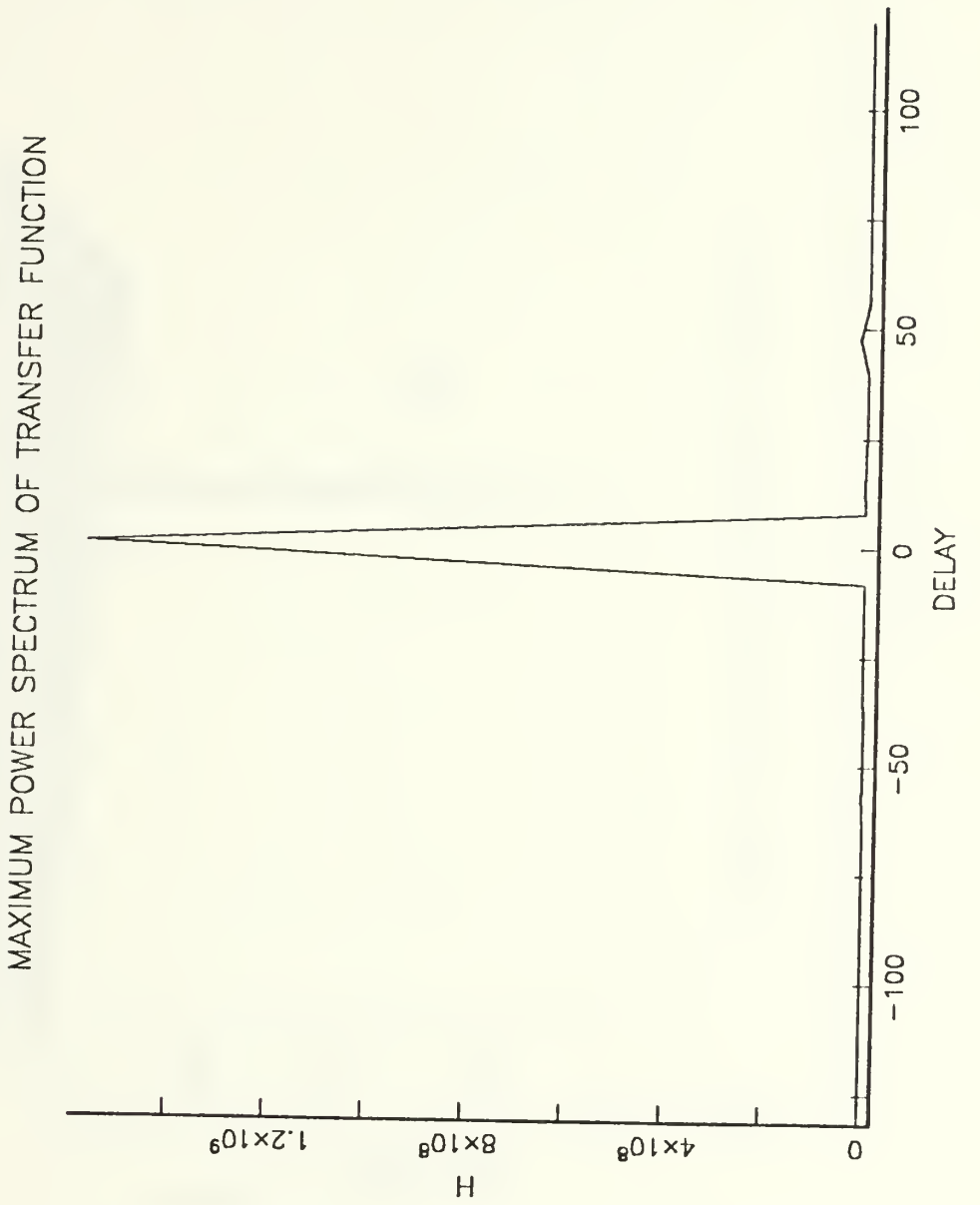

Figure 25. Maximum modified cross power spectrum of the transfer function.

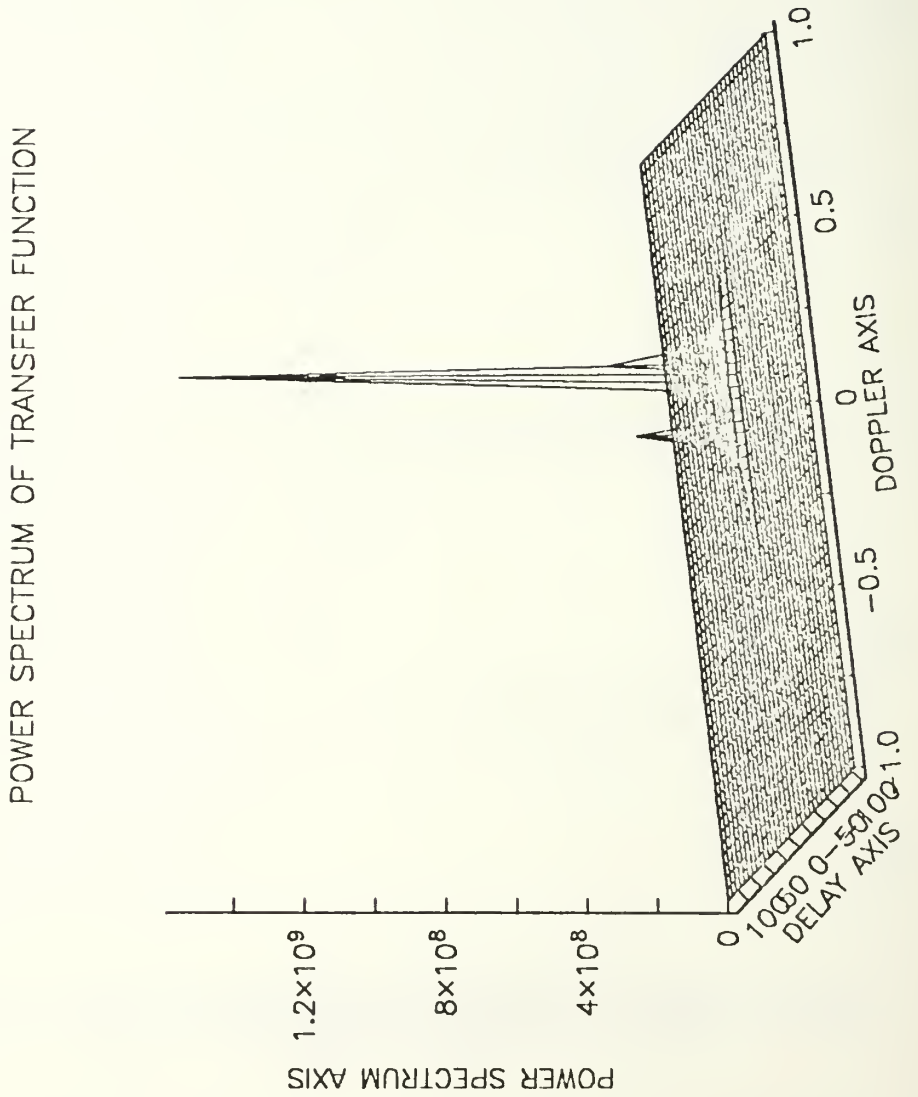

Figure 26. Power spectrum of the transfer function (surface).

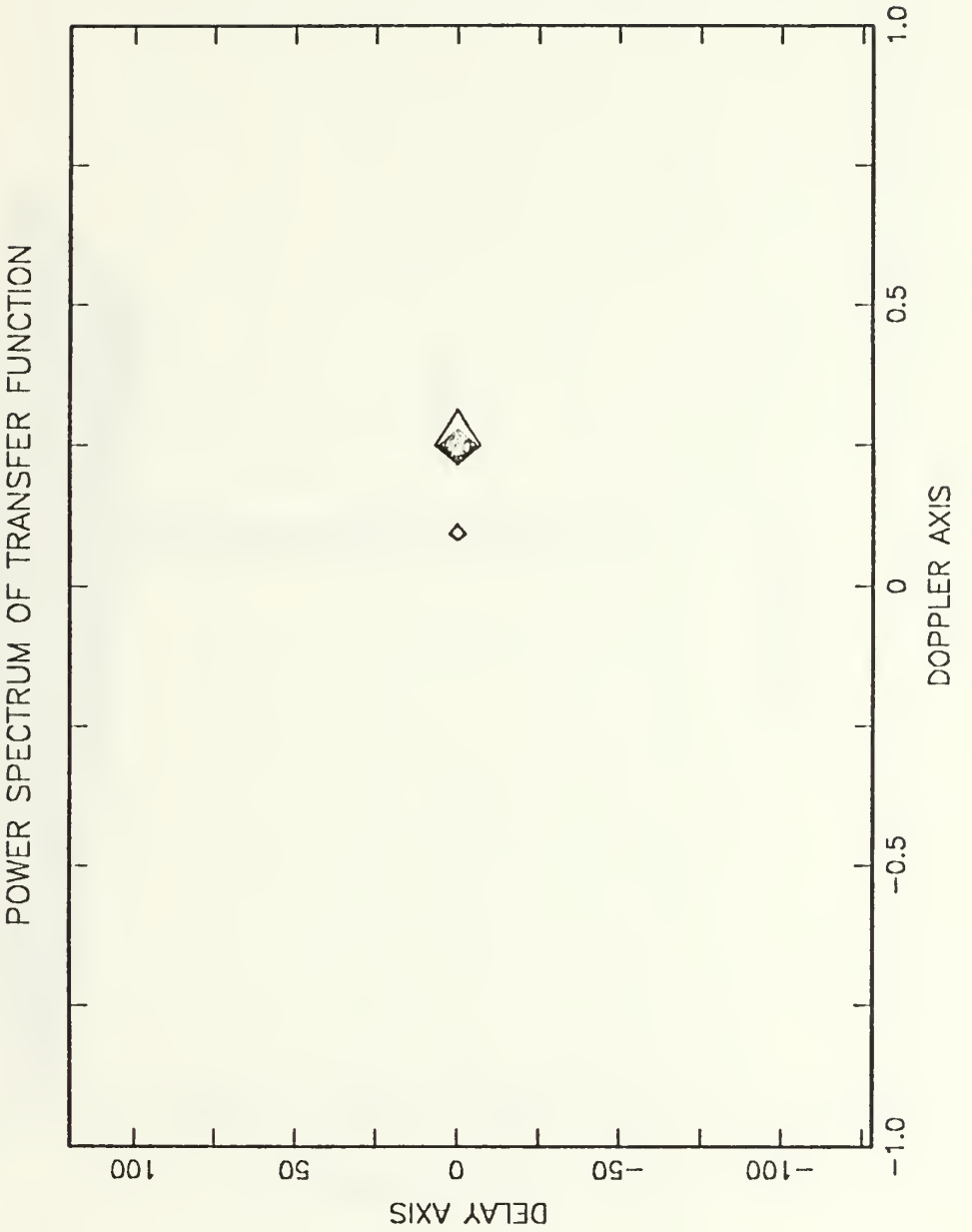

Figure 27. Power spectrum of the transfer function (contour).

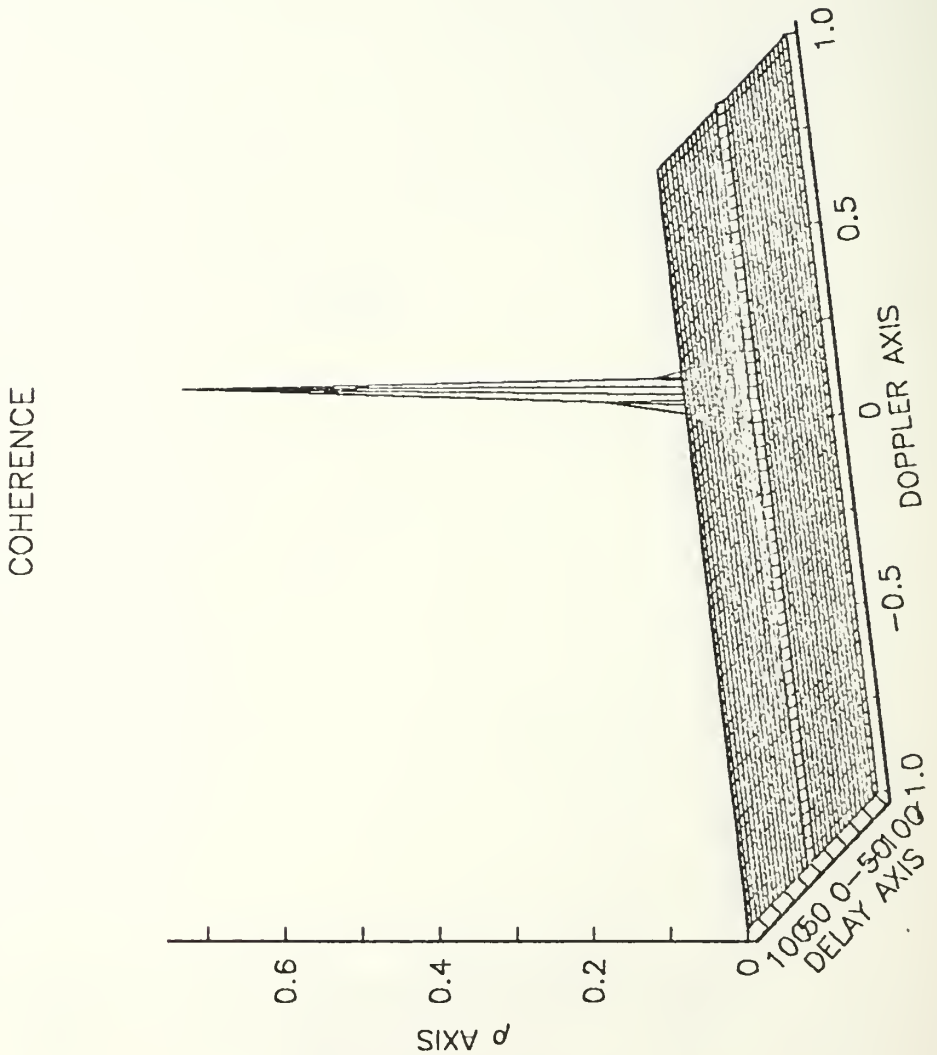

Figure 28. Surface plot of the coherence.

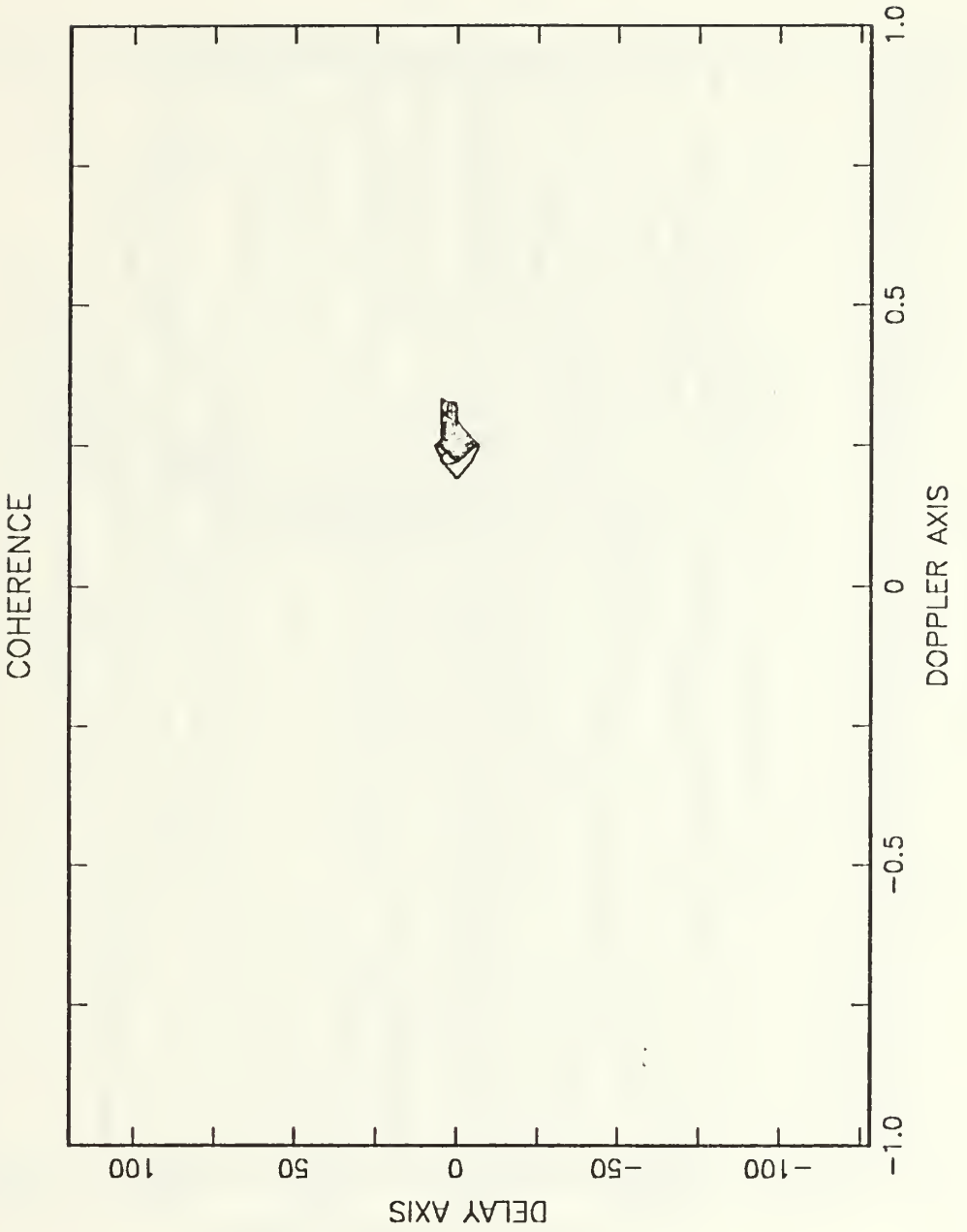

Figure 29. Contour plot of the coherence.

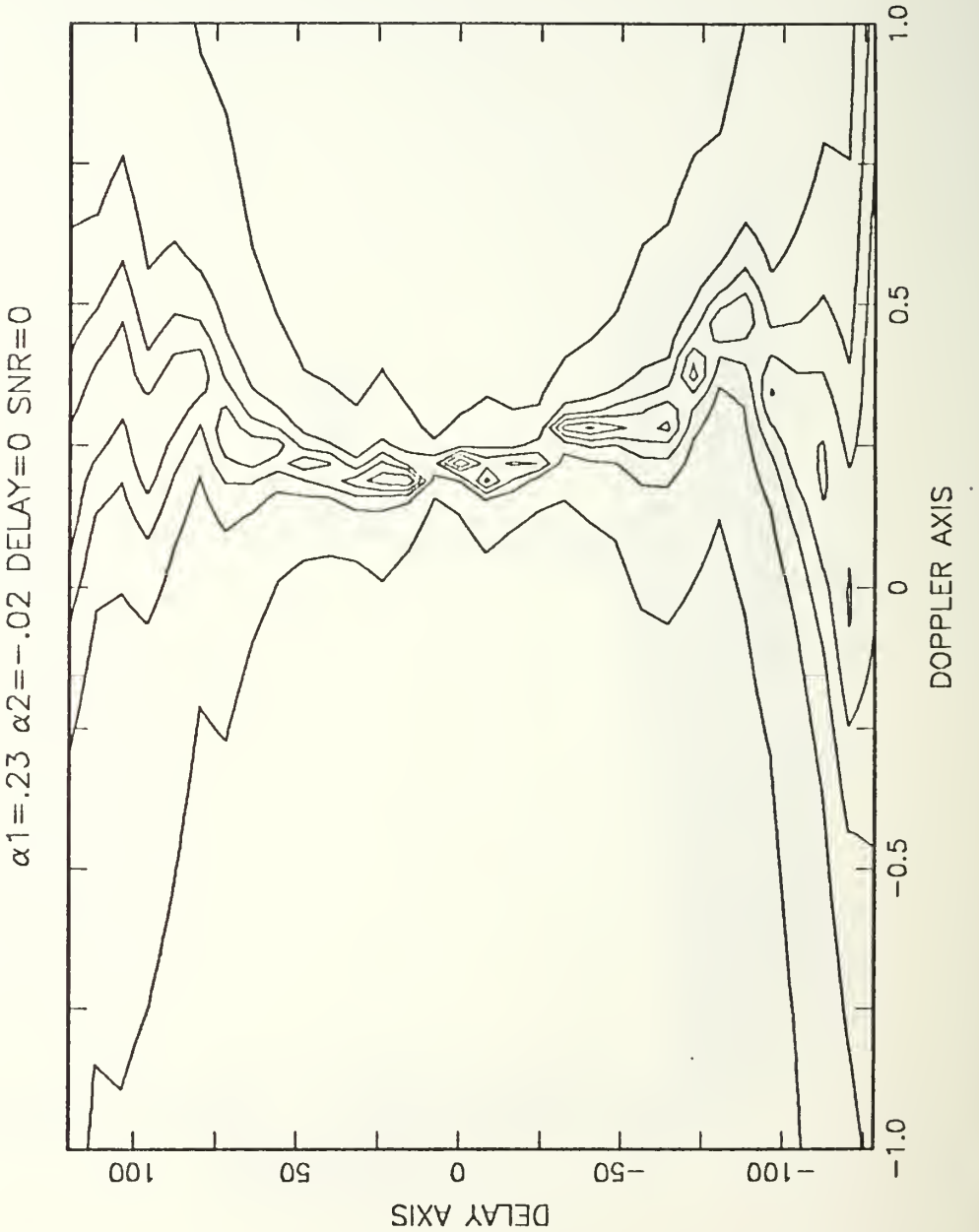

Figure 30. Estimation using the contour shape (low SNR).

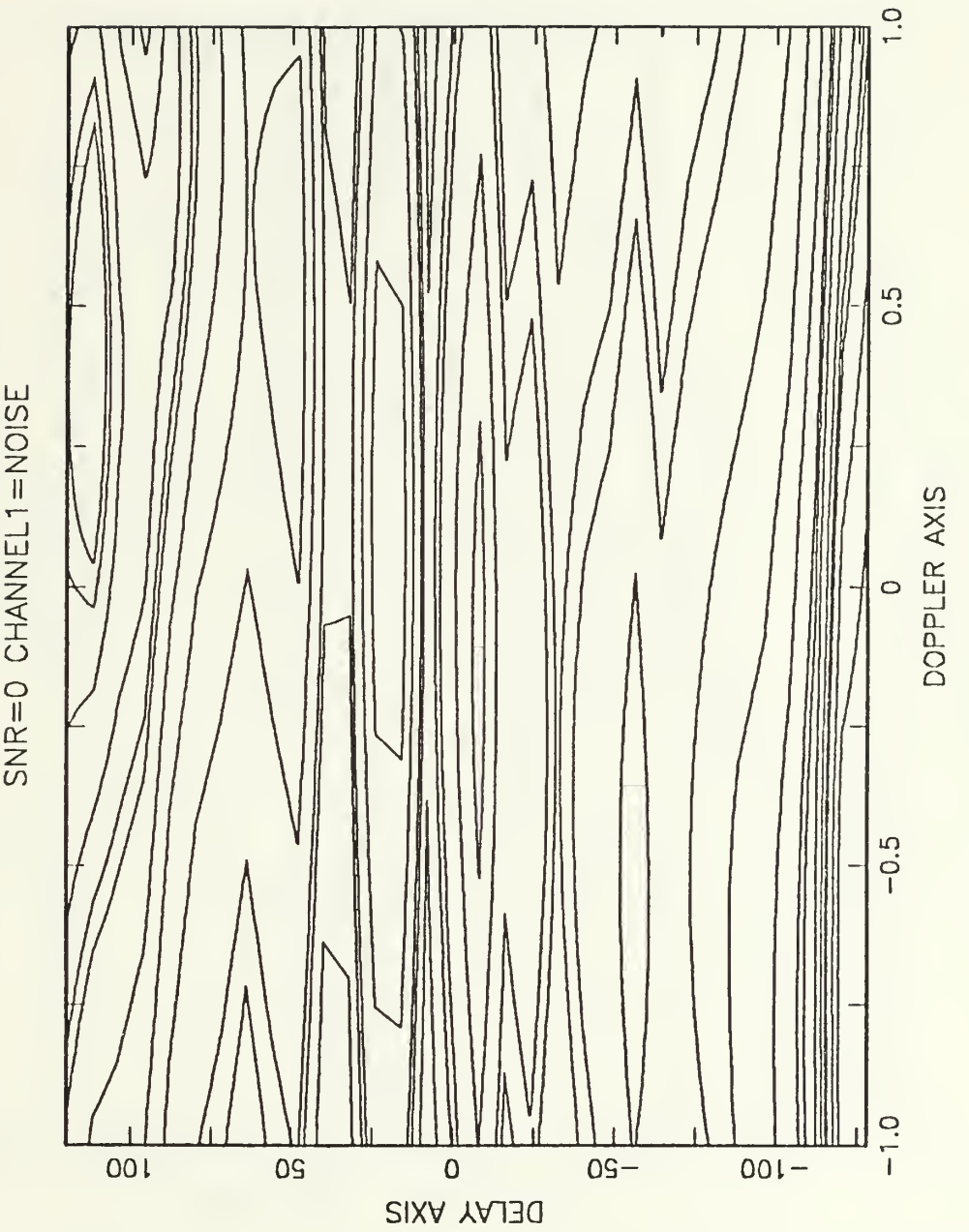

Figure 31. Contour plot (noise in channel-1 only).

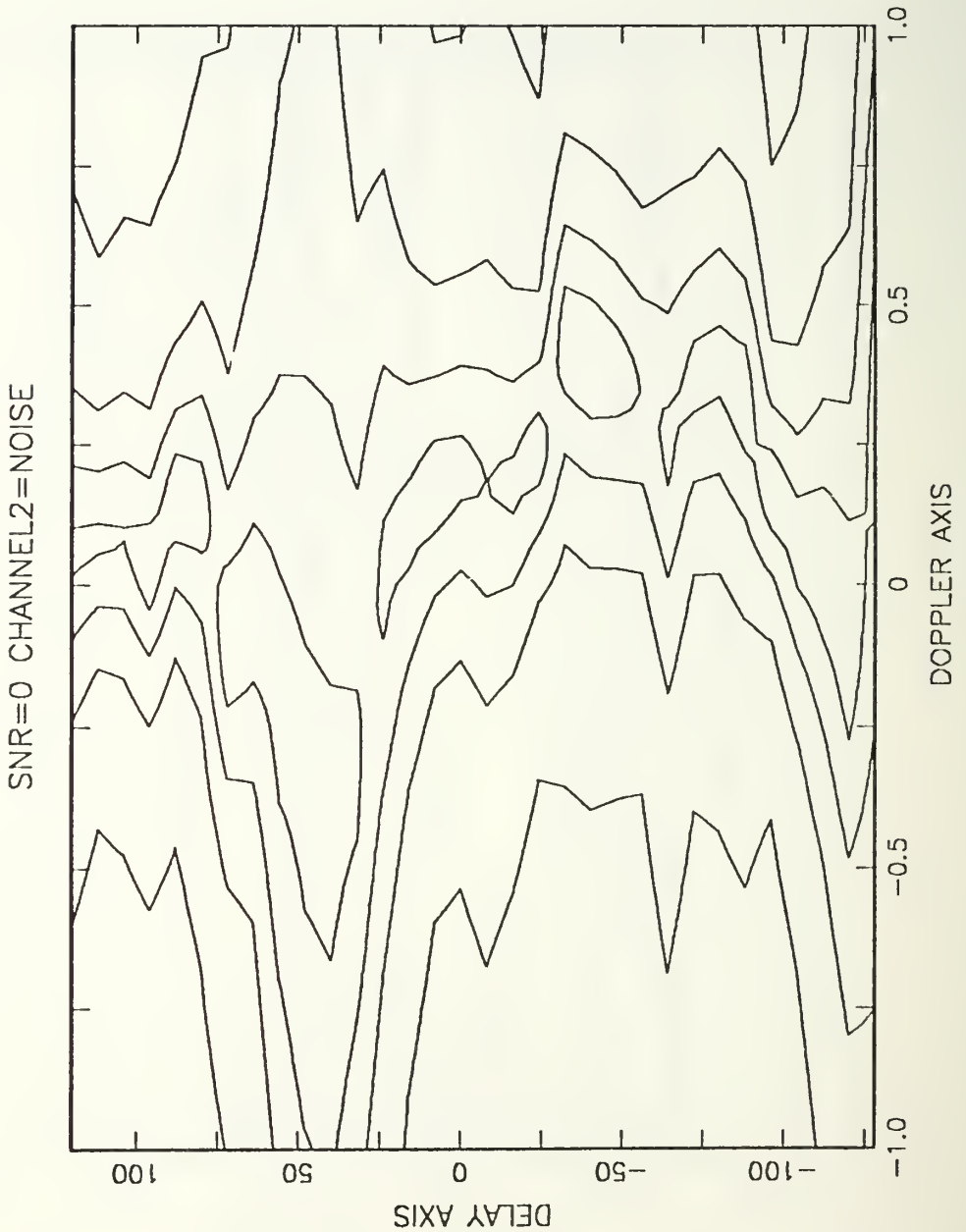

Figure 32. Contour plot (noise in channel-2 only).

Figure 33. Contour plot (noise only).

VI. CONCLUSIONS AND RECOMMENDATIONS

In this thesis, the differential time delay and differential Doppler are estimated using two different approaches. The first approach uses a modified cross power spectrum algorithm while the second approach uses a coherence algorithm. In both cases autoregressive modeling of the required cross spectral component is used. The differential time delay and Doppler difference can be estimated using a threshold at high *SNRs* and a contour plot at low *SNRs*.

At high *SNRs* (i.e., $SNR \geq 70$ dB), the first approach locates the dominant peak at the proper time delay and the proper Doppler. The second approach utilizes two additional AR models and two additional FFTs to obtain the autopower spectrum of channel-1 and channel-2. At moderate *SNRs* (i.e., $SNR \geq 20$ dB), the second approach has the highest coherence peak at the proper time delay and Doppler frequency. When the $SNR \geq 0$ dB, the contour plot of the coherence has the shape of the letter X. The differential time delay and Doppler difference can be located at the cross over point of the X. In high *SNRs* we can use the highest peak to detect the time delay and Doppler frequency using either approach. When we use the contour plot, the information about time delay and Doppler depends on the location and the form of the cross over point of the X.

We can estimate the differential time delay and differential Doppler using the AR modeling of coherence. But we can not get the numerically correct value of the coherence coefficient. The following three points are recommended for future study.

1. To get a cross power spectrum, we assumed that $|P_{xy_1y}(f)| \approx |P_{xy}(f)|^2$. This was required since we can not get a cross power spectrum using an AR model. Improvement of the cross power spectrum estimates using an AR model can lead to improvement of the algorithm.
2. Our first approach is not normalized by the auto power spectra. If we find the corresponding auto power spectra, we can also obtain a coherence coefficient estimate.
3. To get a better information at a low *SNR*, we need to study the characteristics of the contour plots.

APPENDIX A. PHASE DERIVATION

Define the phase at instant k.

$$ {}_x\phi_k = \frac{2\pi(f+\alpha_1)k}{f_s} \tag{4.4} $$

$$ {}_y\phi_k = \frac{2\pi(f+\alpha_2)k}{f_s} \tag{4.5} $$

Let $x(n)$ denote the sampled analog signal $x(t)|_{t=nT}$
then

$$ \begin{aligned} x(n) &= \cos\{2\pi(f+\alpha_1)\frac{n}{f_s}\} \\ &= \cos\{2\pi(f+\alpha_1)\frac{n}{f_s} + {}_x\phi_0\} \end{aligned} \tag{a.1} $$

$$ \begin{aligned} x(n+1) &= \cos\{2\pi(f+\alpha_1)\frac{n+1}{f_s}\} \\ &= \cos\{2\pi(f+\alpha_1)\frac{n}{f_s} + \frac{2\pi(f+\alpha_1)}{f_s}\} \\ &= \cos\{2\pi(f+\alpha_1)\frac{n}{f_s} + {}_x\phi_1\} \end{aligned} \tag{a.2} $$

$$ \begin{aligned} x(n+2) &= \cos\{2\pi(f+\alpha_1)\frac{n+2}{f_s}\} \\ &= \cos\{2\pi(f+\alpha_1)\frac{n}{f_s} + \frac{2\pi(f+\alpha_1)2}{f_s}\} \\ &= \cos\{2\pi(f+\alpha_1)\frac{n}{f_s} + {}_x\phi_2\} \end{aligned} \tag{a.3} $$

$$ \begin{aligned} x(n+m) &= \cos\{2\pi(f+\alpha_1)\frac{n+m}{f_s}\} \\ &= \cos\{2\pi(f+\alpha_1)\frac{n}{f_s} + \frac{2\pi(f+\alpha_1)m}{f_s}\} \\ &= \cos\{2\pi(f+\alpha_1)\frac{n}{f_s} + {}_x\phi_m\} \end{aligned} \tag{4.7} $$

Similiarily

$$
\begin{aligned}
y(n) &= \cos\{2\pi(f+\alpha_2)\frac{n}{f_s}\} \\
&= \cos\{2\pi(f+\alpha_2)\frac{n}{f_s} + {}_y\phi_0\}
\end{aligned}
\tag{a.5}
$$

$$
\begin{aligned}
y(n+1) &= \cos\{2\pi(f+\alpha_2)\frac{n+1}{f_s}\} \\
&= \cos\{2\pi(f+\alpha_2)\frac{n}{f_s} + \frac{2\pi(f+\alpha_2)}{f_s}\} \\
&= \cos\{2\pi(f+\alpha_2)\frac{n}{f_s} + {}_y\phi_2\}
\end{aligned}
\tag{a.6}
$$

$$
\begin{aligned}
y(n+2) &= \cos\{2\pi(f+\alpha_2)\frac{n+2}{f_s}\} \\
&= \cos\{2\pi(f+\alpha_2)\frac{n}{f_s} + \frac{2\pi(f+\alpha_2)2}{f_s}\} \\
&= \cos\{2\pi(f+\alpha_2)\frac{n}{f_s} + {}_y\phi_2\}
\end{aligned}
\tag{a.7}
$$

$$
\begin{aligned}
y(n+m) &= \cos\{2\pi(f+\alpha_2)\frac{n+m}{f_s}\} \\
&= \cos\{2\pi(f+\alpha_2)\frac{n}{f_s} + \frac{2\pi(f+\alpha_2)m}{f_s}\} \\
&= \cos\{2\pi(f+\alpha_2)\frac{n}{f_s} + {}_y\phi_m\}
\end{aligned}
\tag{4.8}
$$

APPENDIX B. OUTPUT OF BPF2

Define two functions depending on k

$$ {}_kA_y = \sum_{n=0}^{N-1-k} \cos(2\pi f_2 \frac{n}{f_s}) e^{j2\pi f \frac{n}{N}} \tag{4.25} $$

$$ {}_kB_y = \sum_{n=0}^{N-1-k} \sin(2\pi f_2 \frac{n}{f_s}) e^{j2\pi f \frac{n}{N}} \tag{4.26} $$

where $f_2 = f + \alpha_2$.

A. FULL INFORMATION CASE (N POINTS)

N points are data points at BPF2 input.

$$
\begin{aligned}
Y_m^*(f) &= \sum_{n=0}^{N-1} \cos(2\pi f_2 \frac{n}{f_s} + {}_y\phi_m) e^{j2\pi f \frac{n}{N}} \\
&= \sum_{n=0}^{N-1} \{ \cos(2\pi f_2 \frac{n}{f_s}) \cos {}_y\phi_m - \sin(2\pi f_2 \frac{n}{f_s}) \sin {}_y\phi_m \} e^{j2\pi f \frac{n}{N}} \\
&= \cos {}_y\phi_m \sum_{n=0}^{N-1} \cos(2\pi f_2 \frac{n}{f_s}) e^{j2\pi \frac{n}{N}} - \sin {}_y\phi_m \sum_{n=0}^{N-1} \sin(2\pi f_2 \frac{n}{f_s}) e^{-j2\pi \frac{n}{N}} \qquad (b.1) \\
&= {}_0A_y \cos {}_y\phi_m - {}_0B_y \sin {}_y\phi_m \\
&= {}_0A_y \frac{e^{j_y\phi_m} + e^{-j_y\phi_m}}{2} - {}_0B_y \frac{e^{j_y\phi_m} - e^{-j_y\phi_m}}{2j} \\
&= \frac{{}_0A_y + j_0B_y}{2} e^{j_y\phi_m} + \frac{{}_0A_y - j_0B_y}{2} e^{-j_y\phi_m}
\end{aligned}
$$

B. PARTIAL INFORMATION CASE (N-1 POINTS)

$N-1$ points are data points at BPF2 input.

$$
\begin{aligned}
Y_m^*(f) &= \sum_{n=0}^{N-1-1} \cos(2\pi f_2 \frac{n}{f_s} + {}_y\phi_m) e^{j2\pi f \frac{n}{N}} \\
&= \sum_{n=0}^{N-1-1} \{ \cos(2\pi f_2 \frac{n}{f_s}) \cos {}_y\phi_m - \sin(2\pi f_2 \frac{n}{f_s}) \sin {}_y\phi_m \} e^{j2\pi f \frac{n}{N}} \\
&= \cos {}_y\phi_m \sum_{n=0}^{N-1-1} \cos(2\pi f_2 \frac{n}{f_s}) e^{j2\pi \frac{n}{N}} - \sin {}_y\phi_m \sum_{n=0}^{N-1-1} \sin(2\pi f_2 \frac{n}{f_s}) e^{-j2\pi \frac{n}{N}} \qquad (b.2) \\
&= {}_1A_y \cos {}_y\phi_m - {}_1B_y \sin {}_y\phi_m \\
&= {}_1A_y \frac{e^{j_y\phi_m} + e^{-j_y\phi_m}}{2} - {}_1B_y \frac{e^{j_y\phi_m} - e^{-j_y\phi_m}}{2j} \\
&= \frac{{}_1A_y + j_1B_y}{2} e^{j_y\phi_m} + \frac{{}_1A_y - j_1B_y}{2} e^{-j_y\phi_m}
\end{aligned}
$$

C. PARTIAL INFORMATION CASE (N-K POINTS)

$N-k$ points are data points at BPF2 input.

$$
\begin{aligned}
Y_m^*(f) &= \sum_{n=0}^{N-1-k} \cos(2\pi f_2 \frac{n}{f_s} + {}_y\phi_m) e^{j2\pi f \frac{n}{N}} \\
&= \sum_{n=0}^{N-1-k} \{ \cos(2\pi f_2 \frac{n}{f_s}) \cos {}_y\phi_m - \sin(2\pi f_2 \frac{n}{f_s}) \sin {}_y\phi_m \} e^{j2\pi f \frac{n}{N}} \\
&= \cos {}_y\phi_m \sum_{n=0}^{N-1-k} \cos(2\pi f_2 \frac{n}{f_s}) e^{j2\pi \frac{n}{N}} - \sin {}_y\phi_m \sum_{n=0}^{N-1-k} \sin(2\pi f_2 \frac{n}{f_s}) e^{-j2\pi \frac{n}{N}} \qquad (b.3) \\
&= {}_kA_y \cos {}_y\phi_m - {}_kB_y \sin {}_y\phi_m \\
&= {}_kA_y \frac{e^{j_y\phi_m} + e^{-j_y\phi_m}}{2} - {}_kB_y \frac{e^{j_y\phi_m} - e^{-j_y\phi_m}}{2j} \\
&= \frac{{}_kA_y + j_kB_y}{2} e^{j_y\phi_m} + \frac{{}_kA_y - j_kB_y}{2} e^{-j_y\phi_m}
\end{aligned}
$$

APPENDIX C. AMPLITUDE COMPARISON

Let C_1 and C_2 be the complex amplitude.

$$C_1 = \frac{1 - e^{j2\pi\alpha}}{1 - e^{j2\pi\alpha\frac{1}{N}}} \tag{c.1}$$

$$C_2 = \frac{1 - e^{-j2\pi\alpha}}{1 - e^{-j2\pi(2f+\alpha)\frac{1}{N}}} \tag{c.2}$$

where $|\alpha| < 0.5$, and $N \geq 2f + 1 > 2(f + \alpha)$. The numerators of $|C_1|$ and $|C_2|$ are same, because the magnitude of the complex conjugate is the same. If we show that $|1 - e^{j2\pi\alpha\frac{1}{N}}| < |1 - e^{-j2\pi(2f+\alpha)\frac{1}{N}}|$, then we can say $|C_1| > |C_2|$. As shown in Figure 34 $|1 - e^{j2\pi\alpha\frac{1}{N}}| = |AI|$ and $|1 - e^{-j2\pi(2f+\alpha)\frac{1}{N}}| = |BI|$.

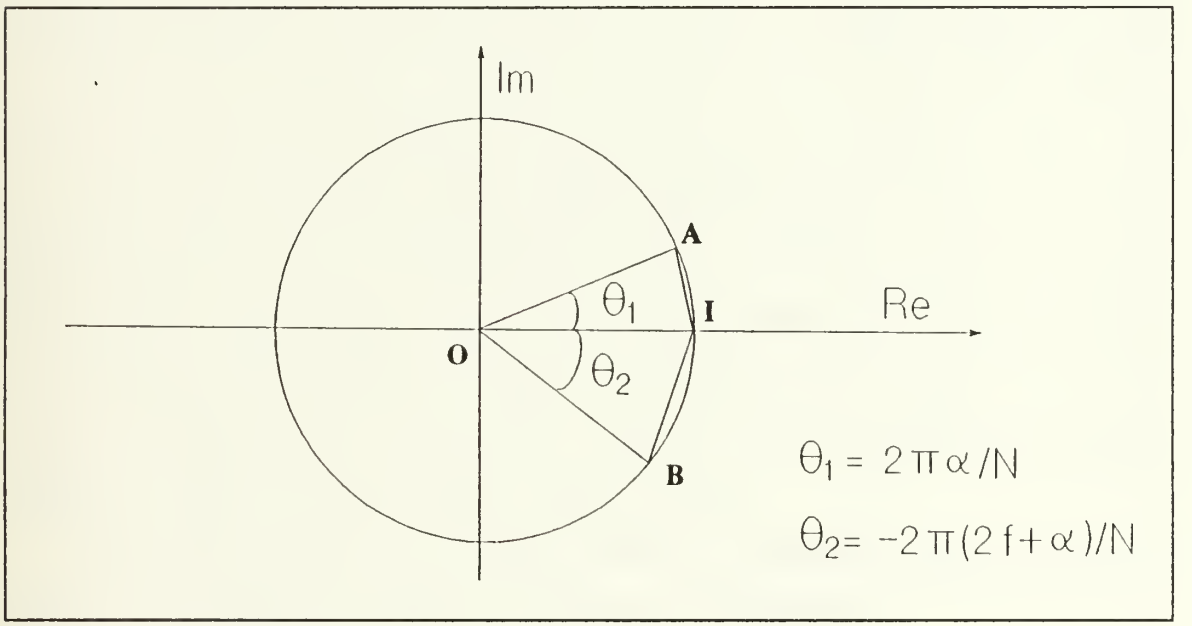

Figure 34. Magnitudes of two complex number.

The smallest angle $\angle IOB$ is $\frac{2\pi(2f+\alpha)}{N}$ or $\frac{2\pi(N-2f-\alpha)}{N} = \frac{2\pi(1-\alpha)}{N}$.

For positive f

$$\angle IOB = \left| \frac{2\pi(2f+\alpha)}{N} \right| > \left| \frac{2\pi\alpha}{N} \right| = \angle IOA \tag{c.3}$$

Since $|1-\alpha| > |\alpha|$.

$$\angle IOB = \left| \frac{2\pi(1-\alpha)}{N} \right| > \left| \frac{2\pi\alpha}{N} \right| = \angle IOA \qquad (c.4)$$

From Eq. (c.3) and Eq. (c.4) $|AI| < |BI|$, therefore

$$|C_1| > |C_2| \qquad (c.5)$$

APPENDIX D. CONTOUR PLOTS OF MODIFIED CROSS POWER SPECTRUM

For *SNRs* greater than or equal to 10 dB the contour plot of the modified cross power spectrum algorithm has the shape of a boomerang. The shape of the boomerang depends on α_2. Figure 35 and Figure 36 show the contour plot of the modified cross power spectrum. The two figures show the boomerang shape and their different directions. For positive α_2 (case 1) and negative α_2 (case 2) the opening of the boomerang is toward the left and right, respectively. In the both cases, we can estimate the time delay and Doppler frequency using these contour plots. The proper time delay and Doppler difference can be detected by locating the point of symmetry of the boomerang. The Figure 37 shows a contour plot when α_2 is -0.02 *Hz*. Since α_2 is very small, the contour plot looks like a straight line. In this case, the differential time delay and Doppler difference can still be estimated by locating the point of symmetry of the boomerang (i.e., center point). Figure 38 shows the contour plot, when channel-1 contains noise only and channel-2 contains signal plus noise (i.e., $SNR = 10$ dB). Figure 39 shows the contour plot, when channel-2 contains noise only and channel-1 contains signal and noise (i.e., $SNR = 10$ dB). Figure 40 shows the contour plot, when both channels contain noise only. Using the modified cross power spectrum technique, the three types of noise only set ups appear as similiar plots (see Figures 38, 39 and 40). Figure 38 seems to indicate the presence of signals in both channels, while Figures 39 and 40 appear more like noise. Hence, contour plots of the modified cross power spectrum do not allow clear identification. In this case, it will be difficult to distinguish between the noise only situation and a small α_2 situation.

To use the modified cross power spectrum algorithm at low *SNRs* (i.e., $SNR < 70$), we need to investigate the contour plot some more.

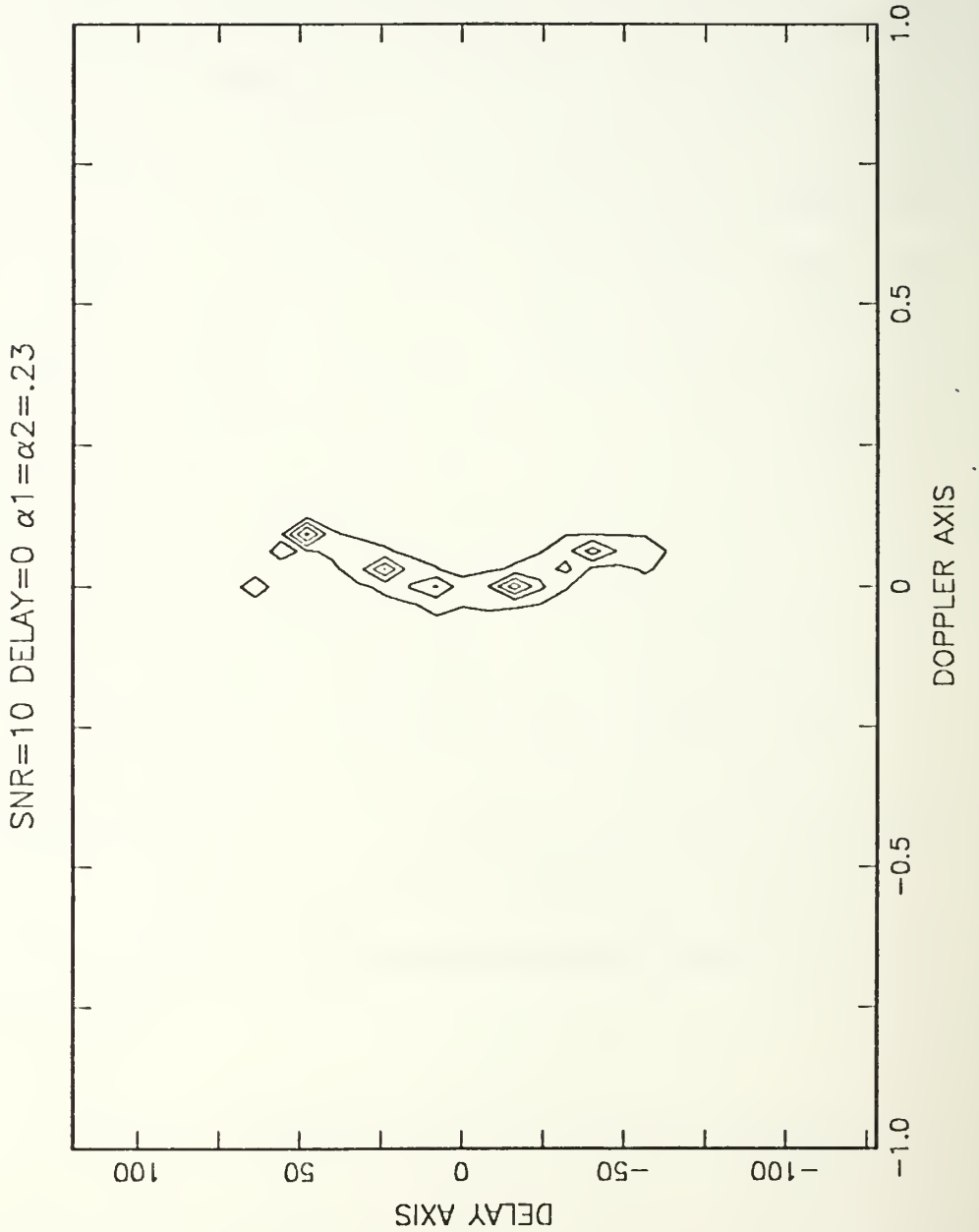

Figure 35. Contour plot of the modified cross power spectrum (case 1).

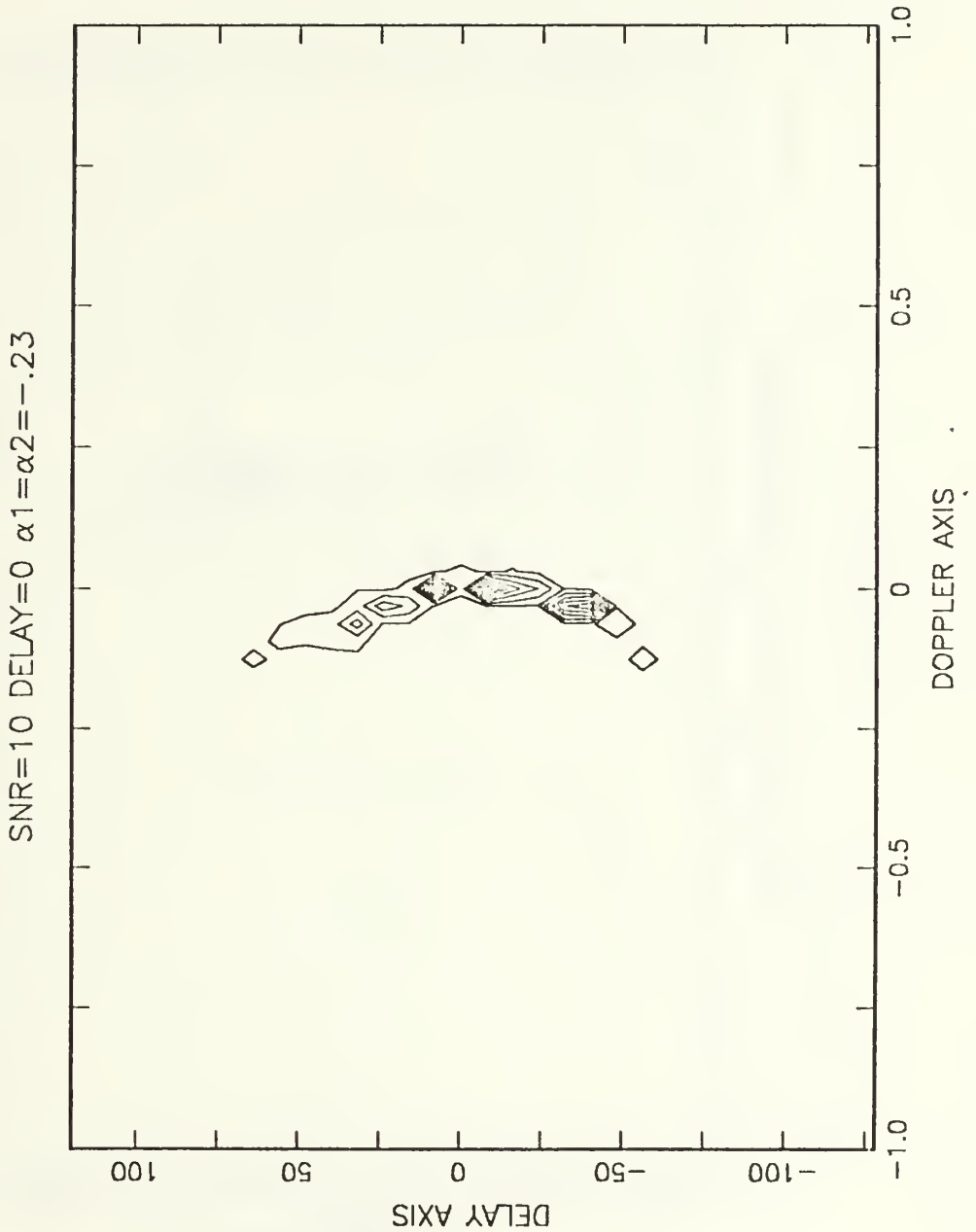

Figure 36. Contour plot of the modified cross power spectrum (case 2).

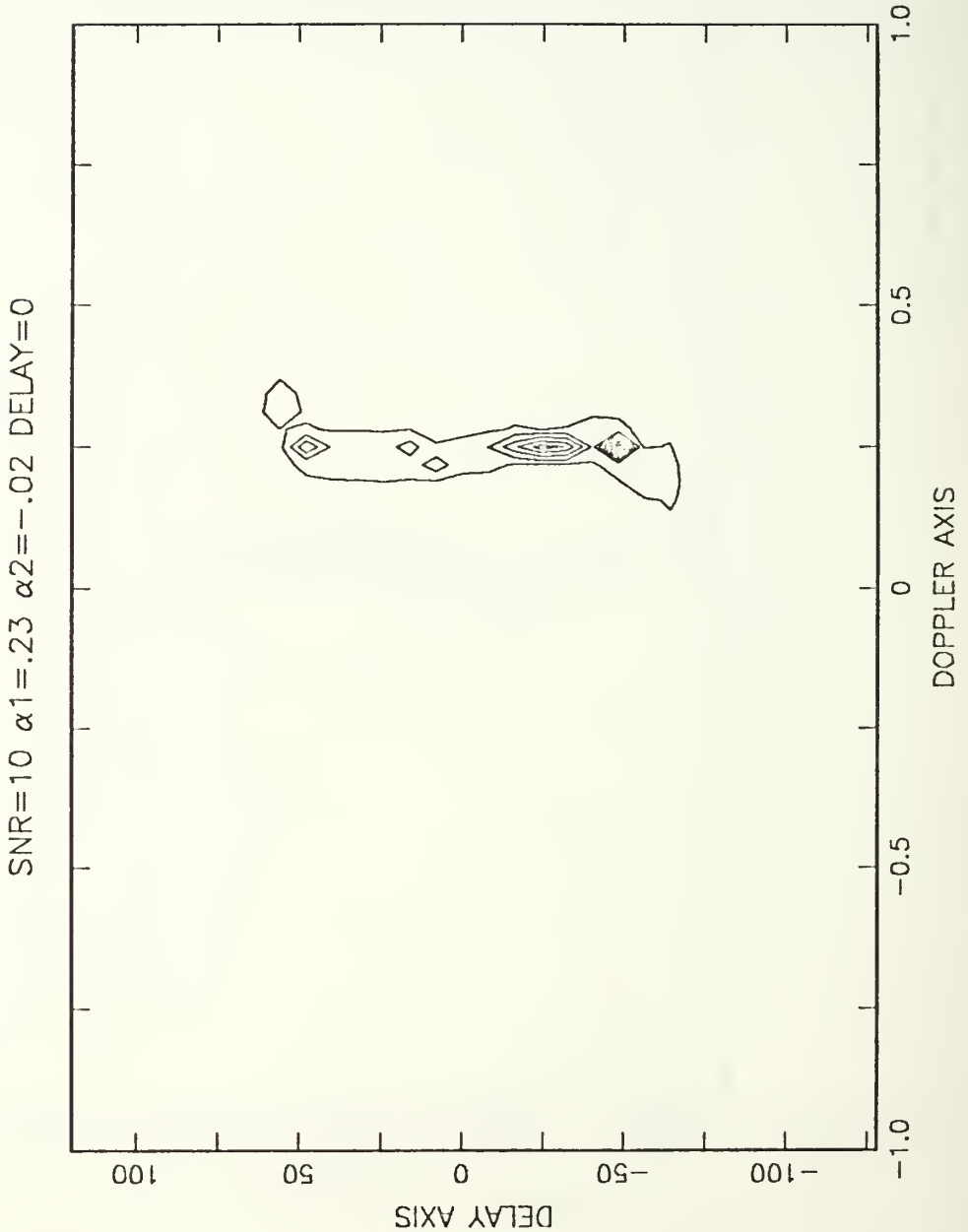

Figure 37. Contour plot of the modified cross power spectrum (case 3).

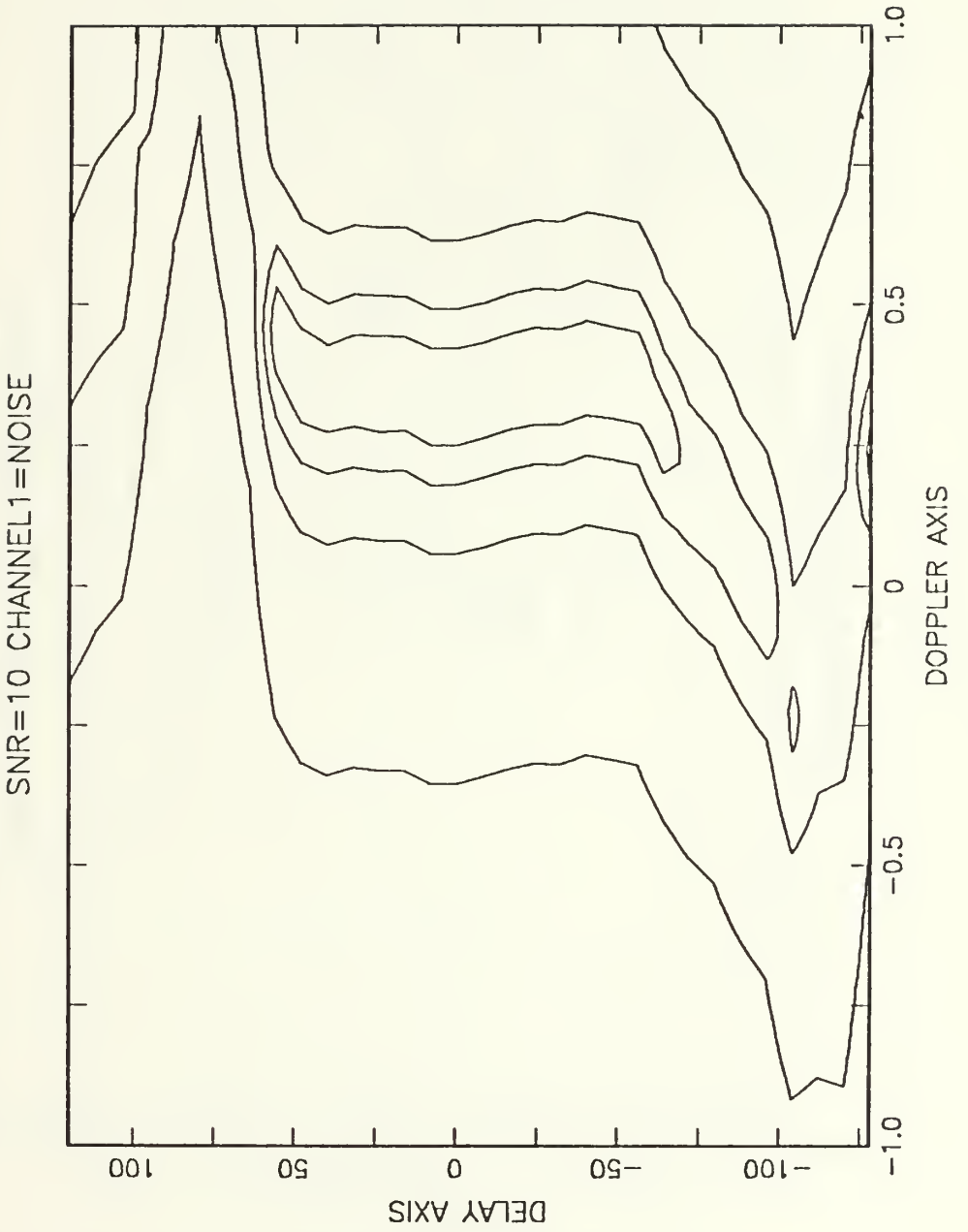

Figure 38. Contour plot of the modified cross power spectrum (noise in channel-1 only).

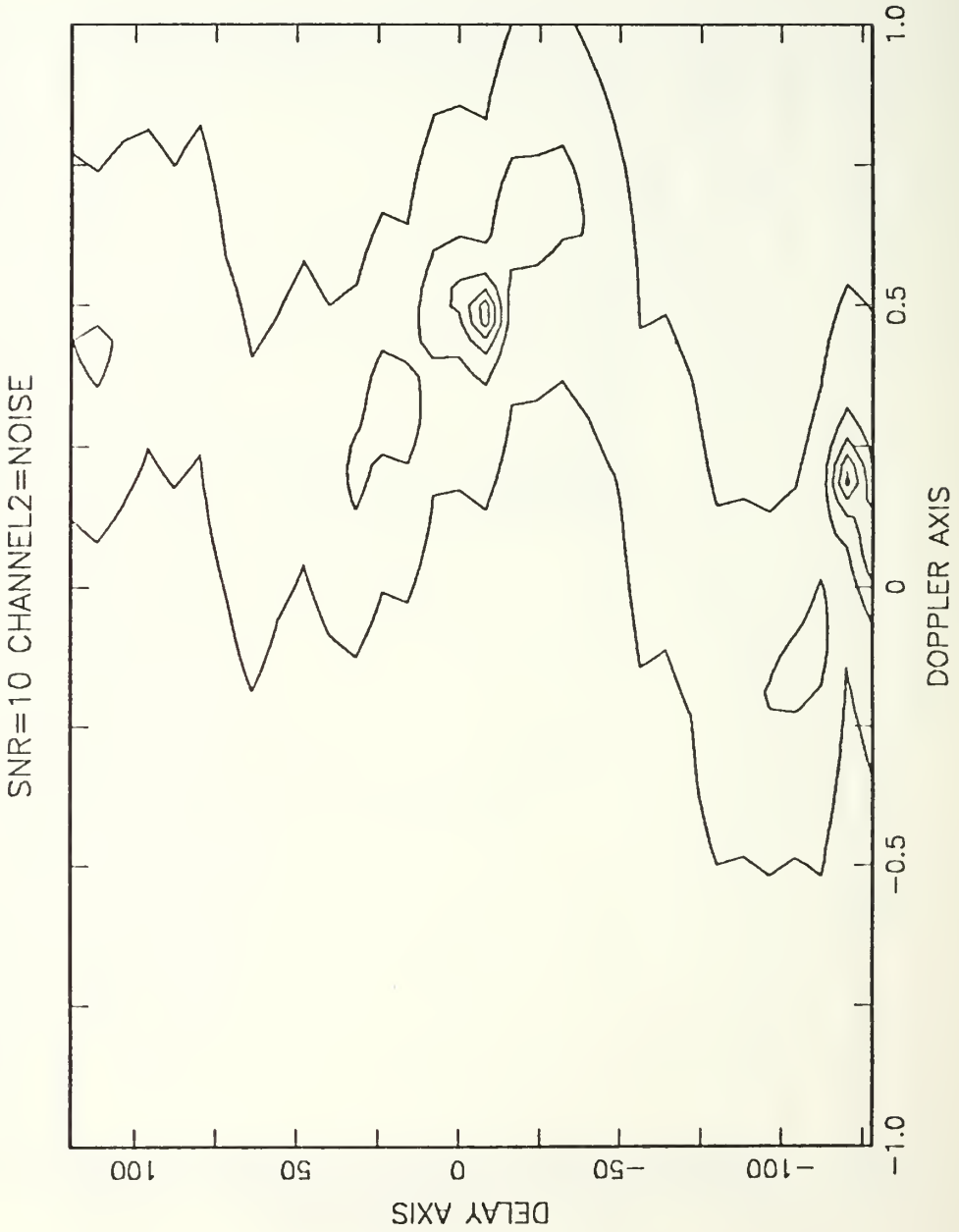

Figure 39. Contour plot of the modified cross power spectrum (noise in channel-2 only).

Figure 40. Contour plot of the modified cross power spectrum (noise only).

APPENDIX E. MODIFIED CROSS POWER SPECTRUM PROGRAM

```
c*****************************************************************************
c                                                                             *
c      main program of approach one.                                          *
c           this program share subroutine with                                *
c           the main program  approach two                                    *
c                                                                             *
c*****************************************************************************
      COMPLEX X(-128:255),Y(-128:255),A(0:2500),B(0:2500)
      COMPLEX BPF1(-128:191),BPF2(-128:191),BPF(0:100)
      INTEGER SNR,BINDEL,DELAY
      DATA X,Y,A,B/5770*(0.,0.)/,BPF1,BPF2,BPF/741*(0.,0.)/
      OPEN(UNIT=7,FILE='FILENAME FILETYPE A')
      PI=ACOS(-1.)
c-----------------------------------------------------------------------------*
c                                                                             *
c     input part                                                              *
c                                                                             *
c     fs       - sampling frequency                                         *
c     f        - carrier frequency                                          *
c     alpha1   - doppler effect at sensor1                                  *
c     alpha2   - doppler effect at sensor2                                  *
c     delay    - signal receiving time difference                           *
c                between two sensors                                        *
c     SNR      - signal to noise ratio                                      *
c     iseed    - noise generator seed(odd number)                           *
c     n        - number of data during one second                           *
c                                                                             *
c-----------------------------------------------------------------------------*
      FS=64.
      F=23.
      ALPHA1=.23
      ALPHA2=-.02
      DELAY=0
      SNR=100
      N=64
      T=2.*PI*F/FS
      ISEED=13
c-----------------------------------------------------------------------------*
c                                                                             *
c        input data sampling at two sensors                                   *
c                                                                             *
c-----------------------------------------------------------------------------*
         CALL SIGNAL(X,F+ALPHA1,FS,0,SNR,ISEED,0,0)
         CALL SIGNAL(Y,F+ALPHA2,FS,DELAY,SNR,ISEED,0,0)
      DO 100 K=-128,191
         DO 50 J=K,K+63
```

```
c-----------------------------------------------------------------------*
c                                                                       *
c          linear transformation using band pass filter                 *
c                                                                       *
c-----------------------------------------------------------------------*
            OMEGA=T*FLOAT(J-K)
                BPF1(K)=BPF1(K)+X(J)*CMPLX(COS(OMEGA),-SIN(OMEGA))
50              BPF2(K)=BPF2(K)+Y(J)*CMPLX(COS(OMEGA),SIN(OMEGA))
c-----------------------------------------------------------------------*
c                                                                       *
c           normalization of BPF2                                       *
c                                                                       *
c-----------------------------------------------------------------------*
100        BPF2(K)=BPF2(K)/CABS(BPF2(K))
      DO 300 BINDEL=-128,127
         DO 200 K=0,N
c-----------------------------------------------------------------------*
c                                                                       *
c           input of AR model                                           *
c                                                                       *
c-----------------------------------------------------------------------*
200                BPF(K)=BPF1(K)*BPF2(K+BINDEL)
         CALL CLEAR(A,B,2500)
         CALL BURGAR(BPF,N+1,A,IP,VAR)
         CALL CHANGEFFT(B,A,11,MAX)
      DO 250 I=-32,-1
250      WRITE(7,997)BINDEL,FLOAT(I)/32.,CABS(1./B(2048+I))*VART
     *,CABS(1./B(2048+I))
      DO 300 I=0,32
300      WRITE(7,997)BINDEL,FLOAT(I)/32.,CABS(1./B(I))*VART
     *,CABS(1./B(I))
      CLOSE(7)
      STOP

997   FORMAT(1X,I4,2X,E13.6,2X,E13.6,2X,E13.6)
      END
```

APPENDIX F. COHERENCE PROGRAM

```
c*************************************************************************************
c                                                                                    *
c     main program of approach two                                                   *
c                                                                                    *
c*************************************************************************************
      COMPLEX X(-128:255),Y(-128:255),A(0:2500),B(0:2500)
      COMPLEX BPF1(-128:191),BPF2(-128:191),BPF(0:100)
      COMPLEX AUTOX(0:100),AUTOY(0:100),SUMXY
      INTEGER SNR,BINDEL,DELAY
      DATA X,Y,A,B/5770*(0.,0.)/,BPF1,BPF2,BPF/741*(0.,0.)/
      DATA AUTOX,AUTOY/202*(0.,0.)/
      OPEN(UNIT=7,FILE='FILENAME FILETYPE A')
      PI=ACOS(-1.)
c-------------------------------------------------------------------------------------*
c                                                                                    *
c     input part                                                                     *
c                                                                                    *
c     fs       - sampling frequency                                                  *
c     f        - carrier frequency                                                   *
c     alpha1   - doppler effect at sensor1                                           *
c     alpha2   - doppler effect at sensor2                                           *
c     delay    - signal receiving time difference                                    *
c                between two sensors                                                 *
c     SNR      - signal to noise ratio                                               *
c     iseed    - noise generator seed(odd number)                                    *
c     n        - number of data during one second                                    *
c                                                                                    *
c-------------------------------------------------------------------------------------*
      ALPHA1=.23
      ALPHA2=-.02
      SNR=20
      F=23.
      N=64
      FS=64.
      T=2.*PI*F/FS
      ISEED=23
      DELAY=0
c-------------------------------------------------------------------------------------*
c                                                                                    *
c     input data sampling at two sensors                                             *
c                                                                                    *
c-------------------------------------------------------------------------------------*
         CALL SIGNAL(X,F+ALPHA1,FS,0,SNR,ISEED,0,0)
         CALL SIGNAL(Y,F+ALPHA2,FS,DELAY,SNR,ISEED,1,0)
      DO 100 K=-128,191
         DO 50 J=K,K+63
c-------------------------------------------------------------------------------------*
c                                                                                    *
c     linear transformation using band pass filter                                   *
c                                                                                    *
c-------------------------------------------------------------------------------------*
```

```
            OMEGA=T*FLOAT(J-K)
                BPF1(K)=BPF1(K)+X(J)*CMPLX(COS(OMEGA),-SIN(OMEGA))
50              BPF2(K)=BPF2(K)+Y(J)*CMPLX(COS(OMEGA),SIN(OMEGA))
100   CONTINUE
      SUMX=0.
      DO 150 K=0,N
c-------------------------------------------------------------------*
c                                                                   *
c     find AR order at sensor1                                      *
c                                                                   *
c-------------------------------------------------------------------*
150      AUTOX(K)=BPF1(K)
         CALL CLEAR(A,B,2500)
         CALL BURGAR(AUTOX,N+1,A,IP,VARX)
         CALL CHANGEFFT(B,A,11,MAX)
              XMAX=CABS(1./B(MAX))
      DO 300 BINDEL=-128,127
         DO 200 K=0,N
c-------------------------------------------------------------------*
c                                                                   *
c     input of AR model                                             *
c                                                                   *
c-------------------------------------------------------------------*
                  AUTOY(K)=BPF2(K+BINDEL)
200               BPF(K)=BPF1(K)*(BPF2(K+BINDEL))
c-------------------------------------------------------------------*
c                                                                   *
c       find auto power spectrum of sensor2 at delay k              *
c                                                                   *
c-------------------------------------------------------------------*
         CALL CLEAR(A,B,2500)
         CALL BURG(AUTOY,N+1,A,IP,VARY)
         CALL CHANGEFFT(B,A,11,MAX)
                     YMAX=1./CABS(B(MAX))
c-------------------------------------------------------------------*
c                                                                   *
c       find auto power spectrum of cross term                      *
c                                                                   *
c-------------------------------------------------------------------*
         CALL CLEAR(A,B,2500)
         CALL BURG(BPF,N+1,A,IP,VARXY)
         CALL CHANGEFFT(B,A,11,MAX)
      XSMAX=(XMAX*YMAX)**2*VARX*VARY/VARXY*fs
      DO 250 I=-32,-1
250   WRITE(7,997)BINDEL,FLOAT(I)/32.,CABS(1./B(2048+I))/SQRT(XSMAX)*8.
     *,CABS(1./B(2048+I))
      DO 300 I=0,32
300      WRITE(7,997)BINDEL,FLOAT(I)/32.,CABS(1./B(I))/SQRT(XSMAX)*8.
     *,CABS(1./B(I))
      CLOSE(7)
      STOP

997   FORMAT(1X,I4,2X,E13.6,2X,E13.6,2X,E13.6)
      END
```

```
C-----------------------------------------------------------------------*
C                                                                       *
C     SIGNAL GENERATER                                                  *
C             INPUT                                                     *
C                     N       - NUMBER OF DATA POINTS                   *
C                     F1,F2   - FREQUENCY OF 1ST,2ND SIGNAL             *
C                     AMP1,AMP2-FREQUENCY OF 1ST,2ND SIGNAL             *
C             OUTPUT                                                    *
C                     A(N)    - SIGANL                                  *
C                                                                       *
C-----------------------------------------------------------------------*
      SUBROUTINE SIGNAL(B,F,FS,D,SNR,ISEED,ID,NOISE)
      COMPLEX B(-128:255)
      REAL A(0:1)
      INTEGER SNR,D
      PI=ACOS(-1.)
      DATA A/0.,0./
      SIGMA=1.
      A(0)=SQRT(2.*10.**(SNR/10))
         IF(NOISE.EQ.1) A(0)=0.
      DO 100 I=-128,255
         M=(I-D)/128
         IF(I.LT.D.OR.M.NE.0) M=1
         T=AMOD(F*FLOAT(I-D),FS)
         CALL GAUSS(ISEED,SIGMA,0.,RANDOM)
         IF(ID.EQ.0) THEN
             X=COS(2.*PI*T/FS)*A(M)+RANDOM
         ELSE
             X=SIN(2.*PI*T/FS)*A(M)+RANDOM
         ENDIF
100      B(I)=CMPLX(X,0.)
      RETURN
      END

C-----------------------------------------------------------------------*
C                                                                       *
C     BURG ALGORITHM                                                    *
C             INPUT                                                     *
C                     N       - NUMBER OF DATA POINTS                   *
C                     X       - INPUT SIGNAL                            *
C             OUTPUT                                                    *
C                     IP      - ORDER OF AR                             *
C                     A(0:IP)- AR COEFFICIENTS                          *
C                     VAR     - DRIVING NOISE VARIATION                 *
C                                                                       *
C-----------------------------------------------------------------------*

      SUBROUTINE BURGAR(X,N,A,IP,VAR)
      COMPLEX X(N),A(0:N),EFK(500),EBK(500)
      COMPLEX EFK1(500),EBK1(500),AA(20,20),SUMN,SUMD
      REAL RHO(0:80),FPE(0:80)
      INTEGER START
      RHO(0)=0.
      FPE(0)=1.E64
      A(0)=CMPLX(1.,0.)
      IP=1
```

```
      START=1
      DO 10 I=1,N
10       RHO(0)=RHO(0)+CABS(X(I))**2/FLOAT(N)
      DO 20 I=2,N
         EFK1(I)=X(I)
20       EBK1(I-1)=X(I-1)
      LOOP
         K=IP
            SUMN=CMPLX(0.,0.)
            SUMD=CMPLX(0.,0.)
            DO 30 I=K+1,N
                 SUMN=SUMN+EFK1(I)*CONJG(EBK1(I-1))
30          SUMD=SUMD+CABS(EFK1(I)**2)+CABS(EBK1(I-1)**2)
C0          SUMD=SUMD+CABS(EFK1(I))**2+CABS(EBK1(I-1))**2
            AA(K,K)=-2.*SUMN/SUMD
         TEMP=1.-CABS(AA(K,K))**2
              IF(TEMP.LE.0.) TEMP=1.E-10
         RHO(K)=TEMP*RHO(K-1)
         IF(K.GT.1) THEN
            DO 40 J=1,K-1
40             AA(J,K)=AA(J,K-1)+AA(K,K)*CONJG(AA(K-J,K-1))
         ENDIF
         DO 60 I=K+2,N
            EFK(I)=EFK1(I)+AA(K,K)*EBK1(I-1)
60          EBK(I-1)=EBK1(I-2)+CONJG(AA(K,K))*EFK1(I-1)
         DO 70 I=K+2,N
            EFK1(I)=EFK(I)
70       EBK1(I-1)=EBK(I-1)
              IF(N-K.EQ.1) THEN
                   FPE(K)=FPE(K-1)+1.
              ELSE
                   FPE(K)=RHO(K)*FLOAT(N+1+K)/FLOAT(N-1-K)
              ENDIF
         IP=IP+1
      UNTIL( FPE(K).GT.FPE(K-1).AND.K.GT.START)
      IP=K-1
      DO 100 I=1,IP
100      A(I)=AA(I,IP)
      VAR=RHO(IP)
      RETURN
      END


C-----------------------------------------------------------------------*
C                                                                       *
C     BURG ALGORITHM                                                    *
C              INPUT                                                    *
C                       N       - NUMBER OF DATA POINTS                 *
C                       X       - INPUT SIGNAL                          *
C                       IP      - ORDER OF AR                           *
C              OUTPUT                                                   *
C                       A(0:IP)- AR COEFFICIENTS                        *
C                       VAR     - DRIVING NOISE VARIATION               *
C                                                                       *
C-----------------------------------------------------------------------*
```

```
      SUBROUTINE BURG(X,N,A,IP,VAR)
      COMPLEX X(N),A(0:N),EFK(500),EBK(500)
      COMPLEX EFK1(500),EBK1(500),AA(20,20),SUMN,SUMD
      REAL RHO(0:80),FPE(0:80)
      INTEGER START
      RHO(0)=0.
      A(0)=CMPLX(1.,0.)
      DO 10 I=1,N
10       RHO(0)=RHO(0)+CABS(X(I))**2/FLOAT(N)
      DO 20 I=2,N
         EFK1(I)=X(I)
20       EBK1(I-1)=X(I-1)
      DO 1000 K=1,IP
            SUMN=CMPLX(0.,0.)
            SUMD=CMPLX(0.,0.)
            DO 30 I=K+1,N
                 SUMN=SUMN+EFK1(I)*CONJG(EBK1(I-1))
30          SUMD=SUMD+CABS(EFK1(I))**2+CABS(EBK1(I-1))**2
            AA(K,K)=-2.*SUMN/SUMD
         TEMP=1.-CABS(AA(K,K))**2
              IF(TEMP.LE.0.) TEMP=1.E-10
         RHO(K)=TEMP*RHO(K-1)
         IF(K.GT.1) THEN
            DO 40 J=1,K-1
40             AA(J,K)=AA(J,K-1)+AA(K,K)*CONJG(AA(K-J,K-1))
         ENDIF
         DO 60 I=K+2,N
            EFK(I)=EFK1(I)+AA(K,K)*EBK1(I-1)
60          EBK(I-1)=EBK1(I-2)+CONJG(AA(K,K))*EFK1(I-1)
         DO 70 I=K+2,N
            EFK1(I)=EFK(I)
70       EBK1(I-1)=EBK(I-1)
1000  CONTINUE
      DO 100 I=1,IP
100      A(I)=AA(I,IP)
      VAR=RHO(IP)
      RETURN
      END
```

```
C-----------------------------------------------------------------------*
C                                                                       *
C     CHANGE FFT                                                        *
C             INPUT                                                     *
C                     B       - AR COEFFICIENTS                       *
C                     A       - TEMPORARY USING ARRAY                 *
C                     ISIZE   - ALOG(FFT DATA POINTS)/ALOG2           *
C             OUTPUT                                                    *
C                     A       - POWER SPECTRUM                        *
C                                                                       *
C-----------------------------------------------------------------------*
```

```
      SUBROUTINE CHANGEFFT(A,B,ISIZE,K)
      COMPLEX A(0:2**ISIZE-1),B(0:2**ISIZE-1)
      INTEGER REVRSE
      PI=ACOS(-1.)
      N=2**ISIZE-1
      K=0
      FS=FLOAT(N+1)
           CALL FFT(N,ISIZE,A,B,FS)
           CALL FINDMAX(A,N,K)
      RETURN
      END

C-------------------------------------------------------------------------*
C                                                                         *
C     FINDMAX                                                             *
C             INPUT                                                       *
C                     N       - NUMBER OF SPECTRUM POINTS(N+1)            *
C                     A       - POWER SPECTRUM                            *
C             OUTPUT                                                      *
C                     MAX     - ARRAY INDEX OF MAXIMUM POWER SPECTRUM     *
C                                                                         *
C-------------------------------------------------------------------------*

      SUBROUTINE FINDMAX(A,N,MAX)
      COMPLEX A(0:N)
      MAX=0
      AMAX=1.E66
      DO 100 I=0,N
         IF(CABS(A(I)).LT.AMAX) THEN
              MAX=I
              AMAX=CABS(A(I))
         ENDIF
100   CONTINUE
      RETURN
      END

C-------------------------------------------------------------------------*
C                                                                         *
C     REVERSE                                                             *
C             BIT REVERSE ORDER INDEX CHANGING SUBROUTINE                 *
C             INPUT                                                       *
C                     N       - ARRAY INDEX                               *
C                     ISIZE   - ALOG(FFT DATA POINTS)/ALOG2               *
C             OUTPUT                                                      *
C                     REVERSE- BIT REVERSE ORDER INDEX                    *
C                                                                         *
C-------------------------------------------------------------------------*

      INTEGER FUNCTION REVRSE(N,ISIZE)
      INTEGER A(20)
      DO 200 I=1,ISIZE
         A(I)=MOD(N,2)
         N=N/2
200   CONTINUE
         REVRSE=0
      DO 300 I=1,ISIZE
```

```
         REVRSE=REVRSE+A(I)*2**(ISIZE-I)
300   CONTINUE
      RETURN
      END

C-------------------------------------------------------------------*
C                                                                   *
C     DFT                                                           *
C         COOLEY - TUKEY ALGORITHM                                  *
C             INPUT                                                 *
C                     N       - NUMBER OF DATA POINTS               *
C                     ISIZE   - ALOG(N)/ALOG2                       *
C                     A       - BIT REVERSE ORDER INDEXED TIME DOMAIN *
C                     B       - TEMPORARY ARRAY                     *
C             OUTPUT                                                *
C                     A       - FREQUENCY DOMAIN                    *
C                                                                   *
C-------------------------------------------------------------------*

      SUBROUTINE DFT(N,ISIZE,A,B)
      COMPLEX A(0:N),B(0:N),W
      PI=ACOS(-1.)
      DO 500 I1=1,ISIZE
         ISAGE1=2**(I1-1)
         ISTAGE=2**I1
         DO 200 I=0,N
              ITEST=MOD(I,ISTAGE)
              IF(ITEST.GT.ISAGE1) THEN
                   K=ITEST-ISAGE1
                        THEATA=2.*PI*FLOAT(K)/FLOAT(ISTAGE)
                        T1=SIN(THEATA)
                        T2=COS(THEATA)
                              IF(ABS(T1).LT.1E-5) T1=0
                              IF(ABS(T2).LT.1E-5) T2=0
                   W=CMPLX(T2,-T1)
                   A(I)=A(I)*W
              ENDIF
200      CONTINUE
         DO 300 I=0,N
              ITEST=MOD(I,ISTAGE)
              IF(ITEST.GE.ISAGE1) THEN
                   B(I)=-A(I)+A(I-ISAGE1)
              ELSE
                   B(I)=A(I)+A(I+ISAGE1)
              ENDIF
300      CONTINUE
         DO 400 I=0,N
              A(I)=B(I)
400      CONTINUE
500   CONTINUE
      RETURN
      END
```

```
C-----------------------------------------------------------------------*
C                                                                       *
C     FFT                                                               *
C          1. INDEX CHANGE                                              *
C          2. USE COOLEY TUKEY ALGORITHM                                *
C             INPUT                                                     *
C                     N       - NUMBER OF DATA POINTS                   *
C                     ISIZE   - ALOG(N)/ALOG2                           *
C                     B       - INPUT(TIME DOMAIN)                      *
C                     FS      - SAMPLING FREQUENCY                      *
C             OUTPUT                                                    *
C                     A       - FREQUENCY DOMAIN                        *
C                                                                       *
C-----------------------------------------------------------------------*

      SUBROUTINE FFT(N,ISIZE,A,B,FS)
      COMPLEX A(0:N),B(0:N)
      INTEGER REVRSE
      DO 100 I=0,N
                  K=I
100      A(REVRSE(K,ISIZE))=B(I)
         CALL DFT(N,ISIZE,A,B)
      DO 200 I=0,N
200      A(I)=A(I)/FS
      RETURN
      END
C-----------------------------------------------------------------------*
C                                                                       *
C     CLEAR                                                             *
C             INPUT                                                     *
C                     N       - NUMBER OF DATA POINTS                   *
C                     A,B     - INPUT arrays                            *
C             OUTPUT                                                    *
C                     A,B     - reinitiallized arrays                   *
C                                                                       *
C-----------------------------------------------------------------------*
      SUBROUTINE CLEAR(A,B,N)
      COMPLEX A(0:N),B(0:N)
         DO 100 I=0,N
100           A(I)=B(I)=CMPLX(0.,0.)
      RETURN
      END
```

LIST OF REFERENCES

1. Carter, G. C., *Time Delay Estimation*, Technical Report 5335, Naval Undersea Systems Center, April 1976

2. Carter, G. C. and Knapp, C. H. and Nuttall, A. H., *Estimation of the Magnitude Squared Coherence Function Via Overlaped Fast Fourier Transform Processing*, IEEE Transactions Audio and Electroacoustics, Vol. AU-21, No.4, pp.337-343, August 1973

3. Marple, S. L., Jr., *DIGITAL SPECTRAL ANALYSIS with Applications*, Prentice Hall, Inc., 1987

4. Orfanidis, S. J., *OPTIMUM SIGNAL PROCESSING. An Introduction*, Macmillan, Inc., 1985

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Department of Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor Ralph Hippenstiel, Code 62Hi Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Professor James Miller, Code 62Mr Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
6.	Naval Ocean System Center Attn. Dr. C. E. Persons, Code 732 San Diego, CA 92152	2
7.	Air Force Central Library Postal Code 151-00 Sindaebang Dong, Kwanak Ku, Seoul Republic of Korea	2
8.	Library of Air Force Academy Postal Code 370-72 Chongwon Gun, Chung Cheong Bug Do, Republic of Korea	2
9.	Lee, Jun Postal Code 370-72 Chongwon Gun, Chung Cheong Bug Do, Republic of Korea	2
10.	Dean of Air Force Academy Postal Code 370-72 Chongwon Gun, Chung Cheong Bug Do, Republic of Korea	1

11. Professor Sug, Young Woo
Postal Code 370-72
Chongwon Gun, Chung Cheong Bug Do,
Republic of Korea — 1

12. Professor Lee, Kyee O
Postal Code 370-72
Chongwon Gun, Chung Cheong Bug Do,
Republic of Korea — 1

13. Choi, Man Soo
SMC #1432
Naval Postgraduate School
Monterey, CA 93943 — 1

Thesis
L41135 Lee
c.1 AR modeling of co-
herence in time delay
and Doppler estimation.